On the Cost of Fixed Partial Match Queries in K-d Trees*

Amalia Duch Gustavo Lau Conrado Martínez†

October 27, 2015

Abstract

Partial match queries constitute the most basic type of associative queries in multidimensional data structures such as K-d trees or quadtrees. Given a query $\mathbf{q} = (q_0, \ldots, q_{K-1})$ where s of the coordinates are specified and $K - s$ are left unspecified ($q_i = *$), a partial match search returns the subset of data points $\mathbf{x} = (x_0, \ldots, x_{K-1})$ in the data structure that match the given query, that is, the data points such that $x_i = q_i$ whenever $q_i \neq *$. There exists a wealth of results about the cost of partial match searches in many different multidimensional data structures, but most of these results deal with random queries. Only recently a few papers have begun to investigate the cost of partial match queries with a fixed query $\mathbf{q}$. This paper represents a new contribution in this direction, giving a detailed asymptotic estimate of the expected cost $P_{n,\mathbf{q}}$ for a given fixed query $\mathbf{q}$. From previous results on the cost of partial matches with a fixed query and the ones presented here, a deeper understanding is emerging, uncovering the following functional shape for $P_{n,\mathbf{q}}$

$$P_{n,\mathbf{q}} = \nu \cdot \left(\prod_{i:q_i \text{ is specified}} q_i(1-q_i) \right)^{\alpha/2} \cdot n^{\alpha} + \text{l.o.t.}^{1}$$

in many multidimensional data structures, which differ only in the exponent α and the constant ν, both dependent on s and K, and, for some data structures, on the whole pattern of specified and unspecified coordinates in $\mathbf{q}$ as well. Although it is tempting to conjecture that this functional shape is "universal", we have shown experimentally that it seems not to be true for a variant of K-d trees called squarish K-d trees.

*This work has been partially supported by funds from the Spanish Ministry for Economy and Competitiveness (MINECO), the European Union (FEDER funds) under grant COMMAS (ref. TIN2013-46181-C2-1-R), and the Catalan Agency for Management of Research and University Grants (AGAUR) grant SGR 2014:1034 (ALBCOM).

†Department of Computer Science, Universitat Politècnica de Catalunya. Jordi Girona, 1–3, E-08034, Barcelona, Spain. {duch,glau,conrado}@cs.upc.edu.

[1] Lower order terms, throughout this work.

1 Introduction

Multidimensional data structures have numerous applications in a broad range of areas of Computer Science, like geographical information systems, computer graphics, video games, databases, and data mining just to name a few. General-purpose multidimensional data structures, such as K-dimensional trees (K-d trees, for short) [Ben75] or quadtrees [BF74], must support efficient updates (insertions and deletions) and exact searches, as any ordinary dictionary data structure, but they should also efficiently support *associative queries*: *orthogonal range queries* (which data points in the data structure have all their coordinates within the ranges specified by given lower and upper bounds along each dimension?), *nearest neighbor queries* (which data point in the data structure is closest —according to a fixed given distance measure— to a given query point?), and others. Among these, *partial match* (PM) queries constitute the most basic type of associative query: given a query $\mathbf{q} = (q_0, \ldots, q_{K-1})$ where only s, $0 < s < K$, out of the K coordinates are specified, find the K-dimensional data points $\mathbf{x}$ that match $\mathbf{q}$, that is, the data points $\mathbf{x} = (x_0, \ldots, x_{K-1})$ in the data structure such that $x_i = q_i$ whenever q_i is specified. Besides their intrinsic interest, PM queries are a fundamental associative query since the performance of many other associative queries is closely related to the performance of PM searches [DM02, CDZC01].

There is a vast literature on the analysis of the performance of partial match search[2] in various multidimensional data structures, starting with the seminal paper by Flajolet and Puech [FP86] that gives the expected performance of a random PM search for several multidimensional data structures. As in [FP86], most of the existing literature only covers the performance of PM search when the query is random, that is, when the query is drawn at random with the same distribution as the data points, with $K - s$ of its coordinates "marked" as unspecified. Here, as in the rest of the literature, the cost of partial match searches will be measured by the number of visited nodes of the tree.

In particular, Flajolet and Puech proved the theorem below for random PM search in random standard K-d trees where s coordinates are specified and the pattern of specified and unspecified coordinates is $\mathbf{u}$ (e.g., $\mathbf{u} = SS*$ indicates the first and second coordinates are specified, the third is not).

Theorem 1 (Flajolet, Puech [FP86])**.** *The expected cost $\overline{P}_{n,\mathbf{u}}$ of a random PM search with query pattern $\mathbf{u}$, where s out of the K coordinates of the query are specified and the other $K - s$ are not, in a random standard K-d tree of size n is*

$$\overline{P}_{n,\mathbf{u}} = \beta_{\mathbf{u}} n^{\alpha(s/K)} + l.o.t.,$$

where $\beta_{\mathbf{u}}$ is a constant that depends on the pattern $\mathbf{u}$ and $\alpha(x)$ is the unique real solution in $[0, 1]$ of

$$(\alpha + 2)^x (\alpha + 1)^{1-x} = 2.$$

[2]The algorithm to perform a partial match query is a partial match *search*; however, sometimes we will abuse the terminology and use the term partial match query when we should actually say partial match search.

Similar results have been shown for other variants of K-d trees and other multidimensional data structures. For instance, for relaxed K-d trees (defined in Section 2) we have the following result.

Theorem 2 (Duch *et al.* [DECM98], Martínez *et al.* [MPP01])**.** *The expected cost $\overline{P}_{n,s,K}$ of a random PM query where s out of the K coordinates of the query are specified and the other $K-s$ are not, in a random relaxed K-d tree of size n is*

$$\overline{P}_{n,s,K} = \beta_{s,K} n^{\alpha(s/K)} + l.o.t.,$$

where

$$\alpha := \alpha(x) = \frac{1}{2}\left(\sqrt{9-8x}-1\right),$$

and the constant $\beta_{s,K} = \beta(s/K)$ depends only on the ratio s/K, with

$$\beta(x) = \frac{1}{1-x}\frac{\Gamma(2\alpha+1)}{(\alpha+1)\Gamma(\alpha+1)\alpha^2\Gamma^2(\alpha)}.$$

In the case of squarish K-d trees (defined in next section) we have the following theorem.

Theorem 3 (Devroye *et al.* [DJZC00])**.** *The expected cost $\overline{P}_{n,s,K}$ of a random PM query where s out of the K coordinates of the query are specified and the other $K-s$ are not, in a random squarish K-d tree of size n is*

$$\overline{P}_{n,s,K} = \Theta(n^{\alpha(s/K)}), \qquad \text{with } \alpha(x) = 1-x.$$

Only recently a handful of papers have studied the performance of PM search with given fixed queries. In [CJ11, BNS13], the authors investigate the expectation, variance and limit distribution of PM search in 2-dimensional quadtrees; in [DJM14] the authors analyze the expected performance of PM search in standard and relaxed K-d trees when exactly $s=1$ coordinate is specified. In the mentioned papers it is shown that

$$\mathbb{E}[\mathcal{P}_{n,\mathbf{q}}] \sim \nu_{\mathbf{q}} \cdot (q_i \cdot (1-q_i))^{\alpha/2} \cdot n^{\alpha} + \text{l.o.t.}, \tag{1}$$

where $\mathcal{P}_{n,\mathbf{q}}$ is the cost of a PM search with fixed query $\mathbf{q}$, $q_i \in (0,1)$ is the unique specified coordinate in $\mathbf{q}$, $\alpha = \alpha(1/K)$ is the same exponent as in the expected cost for random PM queries (the exponent is different for the different multidimensional data structures considered) and $\nu_{\mathbf{q}}$ depends only on K and s (e.g., for relaxed trees), or on the pattern $\mathbf{u}$ of the query $\mathbf{q}$ (e.g., for standard K-d trees). When $\nu_{\mathbf{q}} = \nu_{\mathbf{u}(\mathbf{q})}$ depends on the pattern, it also depends on s and K, but these quantities are implicit in the query pattern.

The main contribution of the present paper is the generalization of the analysis for general s in standard and relaxed K-d trees. We will show in Theorems 4 (for relaxed K-d trees) and 7 (for standard K-d trees) in Sections 3 and 4, respectively, that the cost $\mathcal{P}_{n,\mathbf{q}}$ of a PM search with fixed query $\mathbf{q}$ in a random

K-d tree of size n, satisfies

$$\mathbb{E}[\mathcal{P}_{n,\mathbf{q}}] = \nu_{\mathbf{q}} \cdot \left(\prod_{i:q_i \text{ is specified, } q_i \in (0,1)} q_i \cdot (1-q_i) \right)^{\alpha/2} \cdot n^{\alpha} + \text{l.o.t.}, \qquad (2)$$

where $\nu_{\mathbf{q}}$ and $\alpha = \alpha(s/K)$ are explicitly computable constants, depending on s and K and on the multidimensional data structure under consideration. In the case of standard K-d trees the constant $\nu_{\mathbf{q}}$ depends also on the pattern $\mathbf{u}$ (that is, on the way in which specified and unspecified coordinates are distributed in $\mathbf{q}$), whereas for relaxed K-d trees the pattern does not affect the value of $\nu_{\mathbf{q}}$. We will thus write the constant factor as $\nu_{\mathbf{u}}$ for standard K-d trees and $\nu_{s,K}$ for relaxed K-d trees. The exponent α of n in (2) is the same (when all $q_i \in (0,1)$) as in the expected cost of a random PM search, as given in Theorems 1 and 2 above.

The paper is organized as follows: in Section 2 we review the different K-d tree variants that we will investigate, the partial match algorithm, and the probabilistic model for random K-d trees which we need for the subsequent analysis. Section 3 is devoted to the analysis of $P_{n,\mathbf{q}} = \mathbb{E}[\mathcal{P}_{n,\mathbf{q}}]$, in particular, to prove Eq. (2) for random relaxed K-d trees (Theorem 4). In the course of this analysis, we will have to study the situation where several specified coordinates attain their extreme value (say $q_i = 0$ or $q_i = 1$); in particular we prove an intermediate result (Theorem 5) about the expected performance of random PM queries where several coordinates are extreme. The next section addresses the analysis of $P_{n,\mathbf{q}}$ in standard K-d trees (Theorem 7); we focus only on the aspects that are specific to standard K-d trees since most of the proofs are very similar to the corresponding ones for relaxed K-d trees. Section 5 covers the experimental study that we have performed to compare our theoretical findings with the experimental data. Given the asymptotic nature of our formulæ, the experiments help us to validate the theoretical results for moderate values of the input size. Although we have not obtained theoretical results for squarish K-d trees the experiments suggest that an analogous equation to (2) does not hold for this variant of K-d trees. We make a conjecture about the expected cost of PM search with fixed queries in squarish K-d trees which has some sound theoretical basis and is supported by the experimental data. Finally, Section 6 summarizes our findings and conclusions, and describes our lines of on-going and future work in this topic. This paper extends many of the preliminary results presented in [DLM14].

2 K-d trees & partial match queries

Let F be a collection of n multidimensional records, each one endowed with a K-dimensional key $\mathbf{x} = (x_0, \ldots, x_{K-1})$, with coordinate x_i drawn from a totally ordered domain $\mathcal{D}_i$. It is generally assumed, without loss of generality, that no two keys in the collection have the same coordinates in any of the dimensions. For convenience, here we will also assume that, for all $0 \leq i < K$, $\mathcal{D}_i = [0,1]$.

Initially proposed by Bentley [Ben75] a K-dimensional tree storing a collection F of multidimensional records can be defined as follows.

Definition 1 (Bentley [Ben75])**.** *A K-dimensional tree (or K-d tree) T of size $n \geq 0$ is a binary tree such that*

- *it is either empty when $n = 0$, or*
- *its root stores a record with key $\mathbf{x}$ and a discriminant i, $0 \leq i < K$, and the remaining $n - 1$ records are stored in the left and right subtrees of T, say T_L and T_R, in such a way that both T_L and T_R are K-d trees, where for any key $\mathbf{y} \in T_L$, it holds that $y_i < x_i$ and for any key $\mathbf{y}' \in T_R$, it holds that $x_i < y'_i$.*

Any K-d tree of size n induces a partition of the domain $\mathcal{D} = \mathcal{D}_0 \times \cdots \times \mathcal{D}_{K-1}$ into $n + 1$ regions, each corresponding to a leaf (or, equivalently an empty subtree) in the K-d tree.

Let $\langle \mathbf{x}, i \rangle$ denote a node that contains a key $\mathbf{x}$ with discriminant i. The *bounding box* of $\langle \mathbf{x}, i \rangle$ is the region of the space delimited by the leaf in which $\mathbf{x}$ falls when it is initially inserted into the tree. Thus, if the root is $\langle \mathbf{x}, i \rangle$, its bounding box is $[0, 1]^K$, the bounding box of its left subtree is $[0, 1] \times \cdots \times [0, x_i) \times \cdots \times [0, 1]$, and so on.

Different variants of K-d trees have been proposed so far; most only differ by the way in which discriminants are assigned to nodes (this is the case for the variants that we will consider here). In the original or *standard* K-d trees by Bentley [Ben75], the root of the tree (at level 0) gets discriminant 0, its subtrees in the first level get 1, ..., those in the $(K - 1)$-th level get $K - 1$, those in the K-th level get 0, and so on, in a cyclic way. In general, nodes at level i of the trees discriminate by coordinate $i \mod K$. In Figure 1 we show a standard 2-d tree of five nodes together with the partition of the space that it induces. Duch et al. [DECM98] proposed *relaxed* K-d trees, where each node is assigned a random discriminant, uniformly and independently drawn from $\{0, \ldots, K - 1\}$. The *squarish* K-d trees of Devroye et al. [DJZC00] try to achieve a more balanced partition of the space by discriminating along the coordinate for which the bounding box of the node is most elongated.

Because of their definitions, the insertion and exact search algorithms for K-d trees are straightforward, and we will not give the details here. Insertions work identically in the three variants, except in the way discriminants are assigned to new inserted nodes. The exact search algorithm is the same for all variants.

In a *partial match search* we are given a query $\mathbf{q} = (q_0, \ldots, q_{K-1})$ with $q_i \in \mathcal{D}_i \cup \{*\} = [0, 1] \cup \{*\}$. Coordinates such that $q_i \neq *$ are called *specified*, otherwise they are called *unspecified*; we assume that the number s of specified coordinates satisfies $0 < s < K$. Specified coordinates can be *extreme* (if $q_i = 0$ or $q_i = 1$), otherwise we call them *regular*. The goal of the PM search is to retrieve all records in the K-d tree that match the pattern $\mathbf{q}$, that is, the records $\mathbf{x}$ such that $x_i = q_i$ whenever $q_i \neq *$. In Figure 1 we show with a dashed line a PM query $\mathbf{q} = (*, q_1)$.

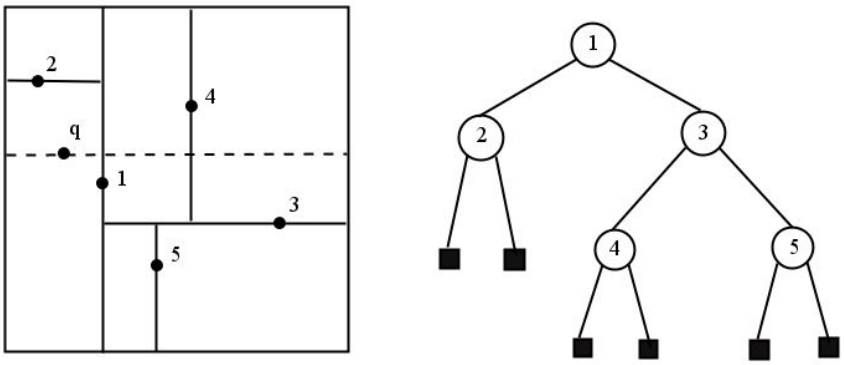

Figure 1: A standard 2-d tree T, the partition that it induces and the PM query $\mathbf{q}$ such that $u(\mathbf{q}) = *S$ and $r(\mathbf{q}, T) = (*, 3)$

The *query pattern*, $\mathbf{u} = u_0\, u_1 \cdots u_{K-1}$, is such that $u_i = S$ if $q_i \neq *$ and $u_i = *$ if $q_i = *$; for queries with extreme specified coordinates, we write $u_i = S$ if the i-th coordinate is regular, and $u_i = E$ if it is extreme. For the query in the example of Figure 1 we have $\mathbf{u}(\mathbf{q}) = *S$.

To perform a PM search with query $\mathbf{q}$, the K-d tree is recursively explored. First, we check whether the root matches $\mathbf{q}$ or not, to report it in the former case. Then, if the root discriminates with respect to an unspecified coordinate, we make recursive calls in both subtrees. Otherwise, if the root is $\langle \mathbf{x}, i \rangle$ we continue recursively in the appropriate subtree, depending on whether $q_i < x_i$ or $q_i \geq x_i$. The exact procedure is shown in Algorithm 1.

The cost of a PM search with query $\mathbf{q}$ in a tree T is measured as the number of nodes of the tree that are visited by Algorithm 1.

We now turn our attention to the probabilistic model that we will use later, when analyzing the expected performance of the previous algorithm.

Definition 2. *A K-d tree T of size n is random if and only if it is either empty ($n = 0$), or if its left and right subtrees, T_L and T_R are random K-d trees of sizes j and $n - 1 - j$, respectively, with*

$$\mathbb{P}[\textit{size of } T_L = j \,|\, \textit{size of } T = n] = \frac{1}{n},$$

for any $0 \leq j < n$.

Another usual characterization of random K-d trees is that they are built by performing n random insertions into an initially empty tree. An insertion is said to be random if the $(j+1)$-th inserted data point $\mathbf{x}$ is equally likely to fall in any of the $j+1$ leaves of the K-d tree; this is what happens if the data points are independently drawn from a vector of continuous distributions in $[0, 1]^K$.

Algorithm 1 PartialMatch reports all records $\mathbf{x}$ in the K-d tree T that match $\mathbf{q}$

```
procedure PartialMatch(q, T)
    ▷ T.discr is the discriminant associated to the root of T
    ▷ T.key is the key associated to the root of T
    ▷ T.left is the left subtree of T
    ▷ T.right is the right subtree of T
    if T = □ then return
    x ← T.key
    if x matches q then
        Report x
    i ← T.discr
    if q_i = * then
        PartialMatch(q, T.left)
        PartialMatch(q, T.right)
    else
        if q_i < x_i then
            PartialMatch(q, T.left)
        else
            PartialMatch(q, T.right)
```

It is worth remembering here that we will say that a PM query $\mathbf{Q}$ is *random* if all its specified coordinates are randomly generated by the same continuous distribution(s) as the coordinates of the keys in the random K-d tree. In other words, the query is random if it is independently generated from the same distribution as the data points, then some of the coordinates are marked "unspecified" as dictated by the query pattern. When all the specified coordinates have fixed values, we say that the query is fixed, i.e., $\mathbf{q}$ is a given element of $\prod_{i=0}^{K-1} \mathcal{D}_i \cup \{*\}$. We will systematically (we have already done that in the introduction) use $\mathcal{P}$ to denote the cost of a PM search, with subscripts for the main parameters: $P_{n,\mathbf{q}} = \mathbb{E}[\mathcal{P}_{n,\mathbf{q}}]$ for the expected cost of a PM search with fixed query $\mathbf{q}$, and $\overline{P}_{n,\bullet} = \mathbb{E}[\mathcal{P}_{n,\mathbf{Q}}]$ for the expected cost of a PM search with a random query $\mathbf{Q}$, the $\bullet$ standing for the other relevant parameters, e.g., the query pattern $\mathbf{u}(\mathbf{Q})$, or the number of specified coordinates s and the dimension K.

In this paper we will assume that each coordinate i, $0 \le i < K$, of each data point is independently drawn from a continuous distribution $\mathcal{F}_i$ in $[0, 1]$. Therefore, a random PM query $\mathbf{Q}$ in a random K-d tree corresponds to an "unsuccessful" PM search, since the probability that $\mathbf{Q}$ matches a key in the tree is zero.

For the sake of simplicity, we can safely assume that $\mathcal{F}_0 = \mathcal{F}_1 = \cdots = \mathcal{F}_{K-1} = \text{Uniform}(0, 1)$. However, for the majority of our results this last assumption is not necessary; we will explicitly state those results where the assumption of uniformity cannot be avoided.

3 Fixed partial match queries in relaxed K-d trees

3.1 Queries and ranks

One key observation about the PM algorithm in K-d trees (the same holds in other comparison-based multidimensional data structures, e.g., quadtrees) is that, except for eventual matches, only the relative ranks of the coordinates matter. To be more precise, let us call the *rank vector* of a query $\mathbf{q}$ the vector $\mathbf{r}(\mathbf{q}, T) = (r_0, \ldots, r_{K-1})$ defined as follows: if $q_i = *$ then $r_i = *$; if $q_i \neq *$ then r_i is the number of records $\mathbf{x}$ in the collection (represented by the K-d tree T) such that $x_i \leq q_i$. Notice that each specified r_i ranges between 0 and n. We will also use the notation $r_k(\mathbf{q}, T)$ to refer to the k-th component of $\mathbf{r}(\mathbf{q}, T)$. For ease of notation, we will usually omit the K-d tree T and simply write $\mathbf{r}(\mathbf{q})$ or $r_k(\mathbf{q})$, unless it is necessary to explicitly indicate which is the tree of reference. In the example of Fig. 1, the rank vector $\mathbf{r}(\mathbf{q}, T) = \mathbf{r}((*, q_1)) = (*, 3)$, since 3 data points have coordinate y smaller that q_1.

Then, for any two given different queries $\mathbf{q}$ and $\mathbf{q}'$ with equal rank vectors $\mathbf{r}(\mathbf{q}) = \mathbf{r}(\mathbf{q}')$, the PM algorithm (Algorithm 1) will visit exactly the same set of nodes of the tree.

It makes sense thus to consider the random variable $\mathcal{P}_{n,\mathbf{r}}$, the cost of a PM query in a random K-d tree of size n where the rank vector of the query is $\mathbf{r}$. The results for $P_{n,\mathbf{r}} = \mathbb{E}[\mathcal{P}_{n,\mathbf{r}}]$ can easily be translated to $P_{n,\mathbf{q}} = \mathbb{E}[\mathcal{P}_{n,\mathbf{q}}]$, via conditioning as follows:

$$P_{n,\mathbf{q}} = \sum_{\mathbf{r}} \mathbb{P}[\mathbf{r}(\mathbf{q}) = \mathbf{r}] \cdot P_{n,\mathbf{r}}.$$

If the coordinates of the records are independently drawn it follows that

$$\mathbb{P}[\mathbf{r}(\mathbf{q}) = \mathbf{r}] = \prod_{q_j \neq *} \mathbb{P}[r_j(\mathbf{q}) = r_j]\,.$$

If we assume that the coordinates are uniformly distributed —here the assumption is a must— then, for $q_j \neq *$, $r_j(\mathbf{q})$ follows a binomial distribution $\mathrm{Bin}(n, q_j)$. Because of the smooth behavior of $P_{n,\mathbf{r}} = \mathbb{E}[\mathcal{P}_{n,\mathbf{r}}]$, and the binomial distribution being highly concentrated around its mean, one can prove that

$$P_{n,\mathbf{q}} = P_{n,\overline{\mathbf{r}}} + \text{l.o.t.}, \tag{3}$$

with $\overline{\mathbf{r}} = n\mathbf{q}$, i.e., $\overline{r}_j = nq_j$ if $q_j \neq *$ and $\overline{r}_j = *$, otherwise. The technical details to finish the proof of (3) are discussed at the end of Subsection 3.6.

Likewise, for a fixed rank vector $\mathbf{r} = (r_0, \ldots, r_{K-1})$, we have

$$P_{n,\mathbf{r}} = P_{n,\overline{\mathbf{q}}} + \text{l.o.t.},$$

with $\overline{q}_i = r_i/n$ if $0 \leq r_i \leq n$ and $\overline{q}_i = *$ if $r_i = *$.

3.2 Main result and outline of its proof

Hereinafter we will concentrate on the analysis of $P_{n,\mathbf{r}}$ since working with ranks conveniently allows a purely combinatorial approach. In this section we will consider the expected cost of a PM search in random relaxed K-d trees, as it turns out to be the multidimensional data structure that is easiest to analyze. The fundamental problems and main steps of the analysis already appear when considering relaxed K-d trees. In Section 4 we are able to extend the analysis to standard K-d trees.

The ultimate goal of our analysis is to find the expected cost $P_{n,\mathbf{q}}$ of a PM search with a fixed query $\mathbf{q}$, that is, to formally prove Equation (2) in the Introduction or, as we have discussed, to solve the essentially equivalent problem of finding the expected cost $P_{n,\mathbf{r}}$ of a PM search with a fixed rank vector $\mathbf{r}$.

It turns out that the distinction between the specified coordinates of a PM query as extreme ($q_i = 0$ or $q_i = 1$) or as regular ($q_i \in (0,1)$) will be of utmost importance in what follows. In terms of rank vectors, a rank r_i will be called extreme if either $r_i = o(n)$ or $r_i = n - o(n)$; it will be called regular otherwise. The number of extreme coordinates s_0, $0 \le s_0 \le s$ (or, equivalently, the number of regular coordinates $t = s - s_0$) will play a fundamental role in the expected cost as we shall see. Namely, we will show that $P_{n,\mathbf{r}}$ is of order n^α, where $\alpha = \alpha(\rho, \rho_0)$, with $\rho = s/K$ and $\rho_0 = s_0/K$. When $\rho_0 = 0$, $\alpha(\rho, 0)$ coincides with the exponent $\alpha(\rho)$ in the expected cost of a random PM query (see Section 1); indeed, the probability that one or more coordinates of a random query are extreme is 0.

Let us now state the main theorem of this section:

Theorem 4. *For a query $\mathbf{q}$ with rank vector $\mathbf{r} = (r_0, \ldots, r_{K-1})$ such that $r_i = z_i n + o(n)$, $0 < z_i < 1$, for all i, $0 \le i < t$, $r_i = o(n)$ or $r_i = n - o(n)$ for all i, $t \le i < s$ and $r_i = *$ for all i, $s \le i < K$, the expected cost of a partial match search in a random relaxed K-d tree of size n is*

$$P_{n,\mathbf{r}} = \nu_{s,t,K} \cdot \left(\prod_{i=0}^{t-1} z_i(1 - z_i) \right)^{\alpha/2} \cdot n^\alpha + o(n^\alpha),$$

where $\rho = s/K$, $\rho_0 = s_0/K = (s-t)/K$, $\alpha = \alpha(\rho, \rho_0)$ is given in Theorem 5 below and $\nu_{s,t,K}$ is:

$$\nu_{s,t,K} = \beta(\rho, \rho_0) \frac{\Gamma^t(\alpha + 2)}{\Gamma^{2t}(\alpha/2 + 1)}.$$

with β also as given in Theorem 5:

$$\beta = \beta(\rho, \rho_0) = \frac{1}{(1-\rho)} \frac{\Gamma(2\alpha + 1 + \rho_0)}{(\alpha+1)\Gamma(\alpha + 1 + \rho_0)\alpha^2\Gamma^2(\alpha)}.$$

Moreover, if the n data points are drawn from the uniform distribution in $[0,1]^K$ then the expected cost of a partial match

$$\mathbf{q} = (q_0, \ldots, q_{t-1}, \overbrace{0 \ldots, 0}^{s_0}, \overbrace{*, \ldots, *}^{K-s})$$

in a random relaxed K-d tree of size n, with $q_i \in (0,1)$ for all $0 \le i < t$, is

$$P_{n,\mathbf{q}} = \nu_{s,t,K} \cdot \left(\prod_{i=0}^{t-1} q_i(1-q_i) \right)^{\alpha/2} \cdot n^\alpha + o(n^\alpha),$$

where $\nu_{s,t,K}$ and α are as above.

In the following subsections we will first analyze the expected cost $\overline{P}_n := \overline{P}_{n,\rho,\rho_0}$ of a PM search with a random query in which s_0 coordinates are extreme and $t = s - s_0$ are chosen at random; the remaining $K - s$ coordinates are left unspecified. The analysis is quite simple and straightforward, but it will give us the analytic closed form for the quantities α and β needed in subsequent steps.

The next step will be to set up the exact recurrence for $P_{n,\mathbf{r}}$. Without loss of generality, for the specific case of relaxed K-d trees we can assume that the first t coordinates of the query (or the rank vector) are regular, then we have s_0 extreme coordinates, then $K - s$ unspecified coordinates. Thus we have $\mathbf{r} = (r_0, r_1, \ldots, r_{t-1}, \overbrace{0, \ldots, 0}^{s_0}, \overbrace{*, \ldots, *}^{K-s})$ and write $\mathbf{r} = (r_0, \ldots, r_{t-1})$. Furthermore we will write $z_i = \lim_{n\to\infty} r_i/n$, $0 \le i < t$, with $0 < z_i < 1$.

In order to solve the (complicated) recurrence for $P_{n,\mathbf{r}}$ and thus prove our main theorem (Theorem 4), we will proceed in several steps: 1) assuming that there is some $\gamma > 0$ such that $\lim_{n\to\infty} n^{-\gamma} P_{n,\mathbf{r}} = f(z_0, \ldots, z_{t-1})$ exists and it is not identically null, we prove that we must have $\gamma = \alpha(\rho, \rho_0)$, and we derive an integral equation, together with a set of boundary/initial conditions, for which $f(z_0, \ldots, z_{t-1})$ is a solution; 2) we solve the integral equation and find the explicit form of $f(z_0, \ldots, z_{t-1})$; 3) once we have a closed form for $f(z_0, \ldots, z_{t-1})$, we explicitly show that

$$f\left(\frac{r_0}{n}, \ldots, \frac{r_{t-1}}{n}\right) n^\alpha + o(n^\alpha)$$

is a solution of the homogeneous recurrence satisfied by $P_{n,\mathbf{r}}$, by unwinding the successive approximations that we made and bounding the corresponding errors. Using the boundary conditions on $P_{n,\mathbf{r}}$, the arbitrary constant in the solution of the integral equation can be fixed to get the desired estimate for $P_{n,\mathbf{r}}$. The last part of Subsection 3.6 is devoted to prove the asymptotic estimate for $P_{n,\mathbf{q}}$, which easily follows from the asymptotic estimate of $P_{n,\mathbf{r}}$ and our discussion in 3.1.

It is also worth mentioning that in the second step above ("solving the integral equation") we will proceed inductively to show that $f(z_0, \ldots, z_{t-1}) = 0$ if any of the z_i's is 0 (or 1).

3.3 Some useful properties of $P_{n,\mathbf{q}}$

The PM algorithm in relaxed K-d trees exhibits some symmetries that can be exploited to simplify the analysis. For instance, the pattern of specified and unspecified coordinates in $\mathbf{q}$ is irrelevant for the expected performance of

the algorithm, and thus we will assume that the query is of the form $\mathbf{q} = (q_0, \ldots, q_{s-1}, \overbrace{*, \ldots, *}^{K-s})$, hence $\mathbf{r}(\mathbf{q}) = (r_0, \ldots, r_{s-1}, \overbrace{*, \ldots, *}^{K-s})$. We will just write $\mathbf{r}(\mathbf{q}) = (r_0, \ldots, r_{s-1})$, omitting the unspecified ranks. We have also $P_{n,\mathbf{r}} = P_{n,\mathbf{r}'}$ if $\mathbf{r}'$ is any permutation of the rank vector $\mathbf{r}$.

In the case of standard K-d trees, where the pattern of specified and unspecified coordinates matters, we have $P_{n,\mathbf{r}} = P_{n,\mathbf{r}'}$ if $\mathbf{r}'$ is any permutation of the specified coordinates of $\mathbf{r}$ (leaving the same unspecified coordinates in both rank vectors). This happens because of the symmetric behavior of left and right subtrees, and the unbiased rule to assign discriminants (no direction is favored). Other "well-behaved" variants of K-d trees will also satisfy $P_{n,\mathbf{r}} = P_{n,\mathbf{r}'}$ for permutations of the specified coordinates.

Another property that applies to any variant of K-d trees is that, because of the symmetry of left and right in K-d trees, we must have that for any i, $0 \leq i < s$,

$$P_{n,(r_0,\ldots,r_{i-1},r_i,r_{i+1},\ldots,r_{s-1})} = P_{n,(r_0,\ldots,r_{i-1},n-r_i,r_{i+1},\ldots,r_{s-1})}. \tag{4}$$

The last useful property that we will use in our analysis is the following: if $\overline{P}_n$ is the expected cost of a random PM query with s, $0 < s < K$, specified coordinates then

$$\overline{P}_n = \frac{1}{(n+1)^s} \sum_{\mathbf{r}} P_{n,\mathbf{r}}. \tag{5}$$

where the summation extends, without loss of generality, over all rank vectors of the form $\mathbf{r} = (r_0, \ldots, r_{s-1}, \overbrace{*, \ldots, *}^{K-s})$ or equivalent.

3.4 Random partial match queries with extreme coordinates

We consider here the expected cost $\overline{P}_n := \overline{P}_{n,\rho,\rho_0}$ of a PM search with a random query with s_0 extreme coordinates, $t = s - s_0$ random coordinates (which will be regular with probability 1), and $K - s$ unspecified coordinates. Our result is cast into the following theorem.

Theorem 5. *Let $\overline{P}_n := \overline{P}_{n,\rho,\rho_0} = \mathbb{E}[\mathcal{P}_{n,\mathbf{Q}}]$ be the expected cost (measured as the number of visited nodes) of a PM search in a random relaxed K-d tree of size n with a random query $\mathbf{Q}$ in which s coordinates are specified and the remaining $K - s$ coordinates are left unspecified. Of the s specified coordinates exactly t coordinates are independently drawn at random from the same continuous distribution(s) from which data coordinates are drawn, and exactly $s_0 = s - t$ coordinates are extreme (we assume, w.l.o.g., that they are 0). Then*

$$\overline{P}_n = \beta(\rho, \rho_0) \cdot n^{\alpha(\rho,\rho_0)} + \textit{l.o.t.},$$

where $\rho = s/K = (s_0+t)/K$, $\rho_0 = s_0/K$,

$$\alpha := \alpha(\rho,\rho_0) = \frac{1}{2}\left(\sqrt{(3-\rho_0)^2 - 8(\rho-\rho_0)} - 1 - \rho_0\right), \quad \textit{and}$$
$$\beta := \beta(\rho,\rho_0) = \frac{1}{(1-\rho)}\frac{\Gamma(2\alpha+1+\rho_0)}{(\alpha+1)\Gamma(\alpha+1+\rho_0)\alpha^2\Gamma^2(\alpha)}.$$

Proof. The proof relies on fairly standard tools of Analytic Combinatorics [FS09], although it involves somewhat lengthy calculations. The first step is to set up a recurrence for $\overline{P}_n$.

If $n = 0$ then we have $\overline{P}_0 = 0$. For $n > 0$, we will condition on the size j of the left subtree; in the probability model that we consider (see Section 2) j is any value in $\{0,\ldots,n-1\}$ with identical probability $1/n$. Now, with probability t/K the root of the random K-d tree of size n discriminates with respect to a specified regular coordinate and the PM search will continue in only one of the subtrees of the random K-d tree; if the size of the subtree is j then the probability that the PM search continues in that subtree is $(j+1)/(n+1)$. With probability s_0/K the root discriminates with respect to an extreme coordinate and the recursion will continue in the left subtree (which is of size j). Finally, with probability $(K-s)/K$ the discriminant at the root corresponds to an unspecified coordinate and the PM search will be called recursively in both subtrees.

Collecting everything and taking into account the symmetries, for $n > 0$

$$\overline{P}_n = 1 + \frac{2t}{K}\frac{1}{n}\sum_{j=0}^{n-1}\frac{j+1}{n+1}\overline{P}_j + \frac{2(K-s)+s_0}{K}\frac{1}{n}\sum_{j=0}^{n-1}\overline{P}_j$$

and then we translate the recurrence into a differential equation for the corresponding generating function. In particular, if $P(z) = \sum_{n\geq 0}\overline{P}_n z^n$ then we obtain the second order linear differential equation

$$P''(z) + P'(z)\frac{2-(4-\rho_0)z}{z(1-z)} - P(z)\frac{4-2\rho-(2-\rho_0)z}{z(1-z)^2} = \frac{2}{z(1-z)^3},$$

with initial conditions $P(0) = \overline{P}_0 = 0$ and $P'(0) = \overline{P}_1 = 1$. The solution is

$$P(z) = \frac{1}{2}\left(\frac{{}_2F_1\left(\begin{smallmatrix}1-\alpha-\rho_0,-\alpha\\2\end{smallmatrix}\,\middle|\, z\right)}{(1-z)^{\alpha+1}} - \frac{1}{1-z}\right),$$

where ${}_2F_1\left(\begin{smallmatrix}a,b\\c\end{smallmatrix}\,\middle|\, z\right)$ denotes the hypergeometric function and

$$\alpha := \alpha(\rho,\rho_0) = \frac{1}{2}\left(\sqrt{(3-\rho_0)^2 - 8(\rho-\rho_0)} - 1 - \rho_0\right).$$

To get the asymptotic estimate for $\overline{P}_n = [z^n]P(z)$ we only need to study the asymptotic behavior of $P(z)$ near its dominant singularity at $z = 1$, and using the transfer lemma of Flajolet and Odlyzko [FO90, FS09] the result follows. □

In the case of relaxed K-d trees, for any ρ and ρ_0 such that $0 \le \rho_0 \le \rho \le 1$, we have $1 - \rho \le \alpha(\rho, \rho_0) \le \alpha(\rho, 0) \le 1$. The most important property to keep now in mind is that the function is decreasing in both ρ and ρ_0. In particular we have

$$1 - \frac{s}{K} = \alpha\left(\frac{s}{K}, \frac{s}{K}\right) < \alpha\left(\frac{s}{K}, \frac{s-1}{K}\right) < \alpha\left(\frac{s}{K}, \frac{s-2}{K}\right) \cdots \\ < \alpha\left(\frac{s}{K}, \frac{1}{K}\right) < \alpha\left(\frac{s}{K}, 0\right) =: \alpha\left(\frac{s}{K}\right).$$

On the other hand, when we take extreme coordinates into consideration Eq. (5) must be changed to

$$\overline{P}_{n,\rho,\rho_0} \sim \frac{1}{(n+1)^t} \sum_{\mathbf{r}} P_{n,\mathbf{r}}, \tag{6}$$

where the summation extends, without loss of generality, over all rank vectors of the form $\mathbf{r} = (r_0, \ldots, r_{t-1}, \overbrace{0, \ldots, 0}^{s_0}, \overbrace{*, \ldots, *}^{K-s})$. In fact, for any rank vector $\mathbf{r}$ of a query of the form $\mathbf{q} = (q_0, q_1, \ldots, q_{t-1}, \overbrace{0, \ldots, 0}^{s_0}, \overbrace{*, \ldots, *}^{K-s})$ with t regular coordinates, s_0 extreme coordinates and $K - s$ unspecified coordinates there are

$$2^{s_0} \binom{K}{t} \binom{K-t}{s_0}$$

rank vectors (including $\mathbf{r}$) with exactly the same expected cost $P_{n,\mathbf{r}}$. If the summation ranged over all possible rank vectors, we would have to divide by

$$(n+1)^s \binom{K}{s}$$

instead. Equation (6) gives only asymptotic equivalence, not equality, since we are considering that the extreme rank values are all identically 0, and disregard the asymptotically negligible contributions of rank vectors with one or more extreme values $r_i = o(n)$.

3.5 Setting up the recurrence

Let us now move to the general recurrence for $P_{n,\mathbf{r}}$. The basis is obviously $P_{0,\mathbf{r}} = 0$. Let $A_{i,j}$ be the event that the root of the random K-d tree T of size n is discriminating with respect to i and that the size of its left subtree L is j, $0 \le j < n$. Then

$$P_{n,\mathbf{r}} = \frac{1}{nK} \sum_{0 \le i < K} \sum_{0 \le j < n} \mathbb{E}[\mathcal{P}_{n,\mathbf{r}} \mid A_{i,j}].$$

We now consider separately two cases—at this point the distinction between regular and extreme specified coordinates is not relevant yet: 1) the discriminating coordinate i of the root is specified ($0 \leq i < s$); 2) it is unspecified ($s \leq i < K$).

Suppose that i is a specified coordinate, that is, $0 \leq i < s$. Now, if $r_i \leq j$ then the PM algorithm will only continue, recursively, in the left subtree, whereas if $j < r_i$ it will only make a recursive call in the right subtree. If, on the other hand, i is an unspecified coordinate then the PM search will proceed recursively into both subtrees. The crucial point is then: how does the rank vector evolve as we "explore" the K-d tree? Let $\mathbf{q}$ be any query such that $\mathbf{r}(\mathbf{q}, T) = \mathbf{r}$. Let $B^{-}_{i,j}(\mathbf{r})$ be the event that i is specified ($0 \leq i < s$), $r_i \leq j$ and $A_{i,j}$. Similarly let $B^{+}_{i,j}(\mathbf{r})$ be the event that i is specified ($0 \leq i < s$), $j < r_i$ and $A_{i,j}$; finally, let $B^{*}_{i,j}(\mathbf{r})$ be the event that i is not specified ($s \leq i < K$) and $A_{i,j}$. We will need to compute the probability $\pi_L^{(i,j)}(\mathbf{r}, \mathbf{r}')$ that $\mathbf{r}(\mathbf{q}, L) = \mathbf{r}'$ given the event $B^{-}_{i,j}(\mathbf{r})$, that is, the probability that $\mathbf{r}(\mathbf{q}, L) = \mathbf{r}'$ when only the left branch is followed. Similarly, we will need to compute the probability $\pi_R^{(i,j)}(\mathbf{r}, \mathbf{r}')$ that $\mathbf{r}(\mathbf{q}, R) = \mathbf{r}'$ given the event $B^{+}_{i,j}(\mathbf{r})$ (with R the right subtree of size $n - 1 - j$ of T); finally, we will also need to compute the probability $\pi_B^{(i,j)}(\mathbf{r}, \mathbf{r}', \mathbf{r}'')$ that $\mathbf{r}(\mathbf{q}, L) = \mathbf{r}'$ and $\mathbf{r}(\mathbf{q}, R) = \mathbf{r}''$ given the event $B^{*}_{i,j}(\mathbf{r})$, that is the probability that the rank vectors are $\mathbf{r}'$ and $\mathbf{r}''$ when both branches must be followed. With these probabilities in hand the recurrence reads

$$
\begin{aligned}
P_{n,\mathbf{r}} = 1 + \frac{1}{nK} \Bigg[\sum_{0 \leq i < s} \Big(& \sum_{j=r_i}^{n-1} \sum_{\mathbf{r}' \in \mathcal{L}_{\mathbf{r}}^{(i,j)}} \pi_L^{(i,j)}(\mathbf{r}, \mathbf{r}') P_{j,\mathbf{r}'} \\
& + \sum_{j=0}^{r_i - 1} \sum_{\mathbf{r}' \in \mathcal{R}_{\mathbf{r}}^{(i,j)}} \pi_R^{(i,j)}(\mathbf{r}, \mathbf{r}') P_{n-1-j,\mathbf{r}'} \Big) \\
& + \sum_{s \leq i < K} \sum_{j=0}^{n-1} \sum_{\langle \mathbf{r}', \mathbf{r}'' \rangle \in \mathcal{B}_{\mathbf{r}}^{(i,j)}} \pi_B^{(i,j)}(\mathbf{r}, \mathbf{r}', \mathbf{r}'') \left(P_{j,\mathbf{r}'} + P_{n-1-j,\mathbf{r}''} \right) \Bigg],
\end{aligned}
$$

where

$$
\begin{aligned}
\mathcal{L}_{\mathbf{r}}^{(i,j)} &= \{ \mathbf{r}' \,|\, r'_i = r_i \wedge \forall k : 0 \leq k < s \wedge k \neq i : 0 \leq r'_k \leq \min(j, r_k) \}, \\
\mathcal{R}_{\mathbf{r}}^{(i,j)} &= \{ \mathbf{r}' \,|\, r'_i = r_i - j - 1 \wedge \forall k : 0 \leq k < s \wedge k \neq i : 0 \leq r'_k \leq \min(n - j - 1, r_k) \}, \\
\mathcal{B}_{\mathbf{r}}^{(i,j)} &= \{ \langle \mathbf{r}', \mathbf{r}'' \rangle \,|\, \mathbf{r}' \in \mathcal{L}_{\mathbf{r}}^{(i,j)} \wedge \mathbf{r}'' \in \mathcal{R}_{\mathbf{r}}^{(i,j)}, \forall k : r_k - 1 \leq r'_k + r''_k \leq r_k \}.
\end{aligned}
$$

If the root discriminates w.r.t. i for some specified coordinate i ($0 \leq i < s$), the size of the left subtree is j and $r_i \leq j$ then the PM search will only continue in the left subtree and $r_i(\mathbf{q}, L) = r_i(\mathbf{q}, T)$. Hence, the rank vectors in the set $\mathcal{L}_{\mathbf{r}}^{(i,j)}$ must all have $r'_i = r_i$. If, on the other hand, $r_i > j$ then the PM search will recursively proceed in the right subtree and $r_i(\mathbf{q}, R) = r_i(\mathbf{q}, T) - j - 1$ since the root and the data points in the left subtree are discarded, hence $r'_i = r_i - j - 1$ for all rank vectors in $\mathcal{R}_{\mathbf{r}}^{(i,j)}$.

In the probabilistic model that we have assumed, the coordinates of each data point are independently drawn and thus

$$\pi_L^{(i,j)}(\mathbf{r},\mathbf{r}') = \prod_{\substack{0\le k<s,\\ k\neq i}} \pi_L^{(i,j)}(k), \quad \pi_R^{(i,j)}(\mathbf{r},\mathbf{r}') = \prod_{\substack{0\le k<s,\\ k\neq i}} \pi_R^{(i,j)}(k),$$
$$\pi_B^{(i,j)}(\mathbf{r},\mathbf{r}',\mathbf{r}'') = \prod_{0\le k<s} \pi_B^{(i,j)}(k),$$

where $\pi_L^{(i,j)}(k)$ is the probability that $r_k(\mathbf{q},L) = r'_k$ given that $r_k(\mathbf{q},T) = r_k$ and the event $B_{i,j}^-(\mathbf{r})$ with $k \neq i$, $\pi_R^{(i,j)}(k)$ is the probability that $r_k(\mathbf{q},R) = r'_k$ given that $r_k(\mathbf{q},T) = r_k$ and the event $B_{i,j}^+(\mathbf{r})$, again for $k \neq i$, and $\pi_B^{(i,j)}(k)$ is the probability that $r_k(\mathbf{q},L) = r'_k$ and $r_k(\mathbf{q},R) = r''_k$ given that $r_k(\mathbf{q},T) = r_k$ and the event $B_{i,j}^*(\mathbf{r})$ (and $k \neq i$ because the i-th coordinate is not specified and k is the index of the specified coordinate).

Let us consider now $\pi_L^{(i,j)}(k)$. It can easily be shown that

$$\pi_L^{(i,j)}(k) = \frac{\binom{j}{r'_k}\binom{n-j}{r_k-r'_k}}{\binom{n}{r_k}}$$

since out of the $\binom{n}{r_k}$ configurations such that $r_k(\mathbf{q},T) = r_k$, the numerator gives the number of configurations in which the left subtree of size j gets r'_k data points where the k-th coordinate is $\le q_k$, and $r_k - r'_k$ of the remaining $n-j$ data points also have their k-th coordinate $\le q_k$. Analogously,

$$\pi_R^{(i,j)}(k) = \frac{\binom{n-1-j}{r'_k}\binom{j+1}{r_k-r'_k}}{\binom{n}{r_k}}.$$

The analysis of $\pi_B^{(i,j)}(k)$ is a bit more complicated, as we must separately consider the situation in which the root has k-th coordinate $\le q_k$ or $> q_k$. If the k-th coordinate of the root is $> q_k$, which happens with probability $(n-r_k)/n$, then

$$\pi_B^{(i,j)}(k) = \frac{\binom{j}{r'_k}\binom{n-1-j}{r_k-r'_k}}{\binom{n-1}{r_k}},$$

that is, we must have $r''_k = r_k - r'_k$, otherwise $\pi_B^{(i,j)}(k) = 0$. If, on the other hand, the k-th coordinate of the root is $\le q_k$, with probability r_k/n, then

$$\pi_B^{(i,j)}(k) = \frac{\binom{j}{r'_k}\binom{n-1-j}{r_k-1-r'_k}}{\binom{n-1}{r_k-1}},$$

hence we must have $r''_k = r_k - 1 - r'_k$, otherwise $\pi_B^{(i,j)}(k) = 0$. This means that in the recurrence the terms corresponding to unspecified coordinates ($s \le i < K$)

can be rewritten as

$$\sum_{s\le i<K}\sum_{j=0}^{n-1}\sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}}\sum_{\boldsymbol{\delta}\in\{0,1\}^s}\prod_{0\le k<s}\mathbb{P}[\Delta_k=\delta_k]\,\frac{\binom{j}{r'_k}\binom{n-1-j}{r_k-\delta_k-r'_k}}{\binom{n-1}{r_k-\delta_k}}\left(P_{j,\mathbf{r}'}+P_{n-1-j,\mathbf{r}-\mathbf{r}'-\boldsymbol{\delta}}\right),$$

where $\boldsymbol{\Delta}$ is the indicator random vector for the event $x_k \le q_k$, and $\mathbf{x}$ is the data point at the root of the random K-d tree.

The complete recurrence can be written in full by gathering all its different "pieces":

$$\begin{aligned}P_{n,\mathbf{r}} = 1+\frac{1}{nK}\Bigg[&\sum_{0\le i<s}\Bigg\{\sum_{j=r_i}^{n-1}\sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}}\prod_{0\le k<s,k\ne i}\frac{\binom{j}{r'_k}\binom{n-1-j}{r_k-r'_k}}{\binom{n-1}{r_k}}P_{j,\mathbf{r}'}\\ &\qquad+\sum_{j=0}^{r_i-1}\sum_{\mathbf{r}'\in\mathcal{R}_{\mathbf{r}}^{(i,j)}}\prod_{0\le k<s,k\ne i}\frac{\binom{n-1-j}{r'_k}\binom{j+1}{r_k-r'_k}}{\binom{n}{r_k}}P_{n-1-j,\mathbf{r}'}\Bigg\}\\ &+\sum_{s\le i<K}\sum_{j=0}^{n-1}\sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}}\\ &\qquad\sum_{\boldsymbol{\delta}\in\{0,1\}^s}\prod_{0\le k<s}h(\delta_k)\frac{\binom{j}{r'_k}\binom{n-1-j}{r_k-\delta_k-r'_k}}{\binom{n-1}{r_k-\delta_k}}\left(P_{j,\mathbf{r}'}+P_{n-1-j,\mathbf{r}-\mathbf{r}'-\boldsymbol{\delta}}\right)\Bigg],\end{aligned}\tag{7}$$

with

$$h(\delta_k)=\mathbb{P}[\Delta_k=\delta_k]=\begin{cases}\frac{r_k}{n}, & \text{if } \delta_k=1,\\ \frac{n-r_k}{n}, & \text{if } \delta_k=0,\end{cases}$$

or more concisely, $h(\delta_k)=(\delta_k r_k+(1-\delta_k)(n-r_k))/n$.

The recurrence looks daunting, and the chances of obtaining the asymptotic behavior of $P_{n,\mathbf{r}}$ as $n\to\infty$ with the standard machinery of Analytic Combinatorics seem extremely small, not to speak of solving the recurrence exactly. But after several simplifications we can obtain a recurrence which is much easier and can be solved using rather standard asymptotic techniques.

3.6 Proof of Theorem 4

Let us recall our strategy to prove Theorem 4. Once we have derived the recurrence that $P_{n,\mathbf{r}}$ satisfies, our next step will be to assume that there is some $\gamma>0$ such that $\lim_{n\to\infty} n^{-\gamma}P_{n,\mathbf{r}}=f(z_0,\dots,z_{t-1})$ exists and it is not zero, which will allow us to prove that we must have $\gamma=\alpha(\rho,\rho_0)$, and to obtain an integral equation, plus a set of boundary/initial conditions, for which $f(z_0,\dots,z_{t-1})$ is a solution. That is, our assumption here—that $\lim_{n\to\infty} n^{-\gamma}P_{n,\mathbf{r}}$ exists and it is not identically null—is just a useful "trick" to transform the original recurrence into the somewhat simpler integral equation (8).

This will be formalized in Proposition 1 below. Then we will solve the integral equation and find the explicit form of $f(z_0,\dots,z_{t-1})$ which is given in

Proposition 2. The rest of this section is mainly devoted to prove that the

$$f\left(\frac{r_0}{n},\ldots,\frac{r_{t-1}}{n}\right)n^\alpha + o(n^\alpha)$$

is a solution of the recurrence satisfied by $P_{n,\mathbf{r}}$.

Last but not least, the asymptotic estimate for $P_{n,\mathbf{r}}$ together with the discussion in Subsection 3.1 allows us to prove the last part of Theorem 4, namely, the asymptotic estimate for $P_{n,\mathbf{q}}$.

Proposition 1. *Let $\mathbf{r} = (r_0,\ldots,r_{K-1})$ be such that $z_i = \lim_{n\to\infty} r_i/n \in (0,1)$ for all i, $0 \le i < t$, $r_i = o(n)$ or $r_i = n - o(n)$ for all i, $t \le i < s$, and $r_i = *$ for all i, $s \le i < K$. Assume that $0 < t \le s < K$, that is, the rank vector corresponds to a partial match query with at least one regular specified coordinate and at least an unspecified coordinate.*

If $\lim_{n\to\infty} \frac{P_{n,\mathbf{r}}}{n^\gamma}$ exists for some γ and it is non-null then $\gamma = \alpha(\rho,\rho_0)$ (as given in Theorem 5) and

$$f(z_0,\ldots,z_{t-1}) = \lim_{n\to\infty}\frac{P_{n,\mathbf{r}}}{n^\gamma}$$

is the solution of the integral equation

$$f(z_0,\ldots,z_{t-1}) = \lambda\sum_{i=0}^{t-1}\left\{ z_i^{\gamma+1}\int_{z_i}^1 f(z_0,\ldots,z_{i-1},z,z_{i+1},\ldots,z_{t-1})\frac{dz}{z^{\gamma+2}} \right. \tag{8}$$
$$\left. + (1-z_i)^{\gamma+1}\int_0^{z_i} f(z_0,\ldots,z_{i-1},z,z_{i+1},\ldots,z_{t-1})\frac{dz}{(1-z)^{\gamma+2}}\right\},$$

where $\lambda = (\alpha+2)/2t$ and the function f is subject to the following constraints:

(a) The function f is symmetric with respect to any permutation of its arguments.

(b) For any i, $0 \le i < t$, and $z_i \in (0,1)$, $f(z_0,\ldots,z_{i-1},z_i,z_{i+1},\ldots,z_{t-1}) = f(z_0,\ldots,z_{i-1},1-z_i,z_{i+1},\ldots,z_{t-1})$.

(c) For any i, $0 \le i < t$,

$$\lim_{z_i\to 0} f(z_0,\ldots,z_{i-1},z_i,z_{i+1},\ldots,z_{t-1}) = \lim_{z_i\to 1} f(z_0,\ldots,z_{i-1},z_i,z_{i+1},\ldots,z_{t-1}) = 0.$$

(d)

$$\int_0^1\cdots\int_0^1 f(y_0,\ldots,y_{t-1})\,dy_0\cdots dy_{t-1} = \beta(\rho,\rho_0).$$

For the proof of the proposition above, we have to go through several approximations of recurrence (7); then plugging the assumption $P_{n,\mathbf{r}}/n^\gamma =$

$f(z_0,\ldots,z_{t-1})+o(1)$ and passing to the limit yields the integral equation (8) for f given in the proposition. The constraints (boundary conditions) that f must satisfy easily follow from the symmetry properties of $P_{n,\mathbf{r}}$. Constraint (d) and $\gamma=\alpha(\rho,\rho_0)$ are consequences of (6). In fact, we must have $\gamma=\alpha(\rho,\rho_0)$, for otherwise we would have a contradiction with our hypothesis: either we would have that $\lim_{n\to\infty} P_{n,\mathbf{r}}/n^{\gamma}=0$ or that limit does not exist (the limit would be ∞). To establish the constraints in (c) we use induction on t, for fixed s; the crucial point here is that $\alpha(\rho,\rho_0)>\alpha(\rho,\rho_0')$ if $\rho_0<\rho_0'$. More details of the proof can be found in Appendix A.

We move next to the second proposition here, giving us the explicit form of the solution to the integral equation (8).

Proposition 2. *Let*

$$f(z_0,\ldots,z_{t-1})=\nu_{s,t,K}\cdot\left(\prod_{i=0}^{t-1} z_i(1-z_i)\right)^{\alpha/2}$$

where $\rho=s/K$, $\rho_0=s_0/K=(s-t)/K$, $\alpha=\alpha(\rho,\rho_0)$ *is given in Theorem 5 and* $\nu_{s,t,K}$ *is*

$$\nu_{s,t,K}=\beta(\rho,\rho_0)\frac{\Gamma^t(\alpha+2)}{\Gamma^{2t}(\alpha/2+1)}.$$

with

$$\beta=\beta(\rho,\rho_0)=\frac{1}{(1-\rho)}\frac{\Gamma(2\alpha+1+\rho_0)}{(\alpha+1)\Gamma(\alpha+1+\rho_0)\alpha^2\Gamma^2(\alpha)}.$$

Then f *is a solution of* (8) *satisfying all the constraints stated in Proposition 1.*

In order to solve the integral equation, we can take successive partial derivatives to obtain a PDE for f. However, because of the symmetries of f on all its arguments, we can assume that

$$f(z_0,\ldots,z_{t-1})=\phi_0(z_0)\cdots\phi_{t-1}(z_{t-1})$$

for some $\phi_0,\ldots,\phi_{t-1}$. Moreover, we can safely assume that $\phi_0=\cdots=\phi_{t-1}=\phi$. So we end up with an ODE for ϕ and a set of constraints on ϕ (e.g., $\phi(z)=\phi(1-z)$ for all $z\in(0,1)$) for which the unique solution can be found quite easily. The complete details of the proof are given in Appendix B.

To complete the proof of Theorem 4 we carefully compute the errors in passing from recurrence (7) to the integral equation (8). Having the explicit form for $f(z_0,\ldots,z_{t-1})$ is crucial here. The details are given in Appendix C, showing that the error is $O(\Delta^2/n)+O(1/n)$ with $\Delta=o(\sqrt{n})$, that is, the error is $o(1)$. The de Moivre-Laplace method that we mentioned above plays also an important role in these computations.

Let us now write recurrence (7) in a more compact form:

$$P_{n,\mathbf{r}}=1+\frac{1}{nK}\sum_{i=0}^{K-1}\sum_{j=0}^{n-1}\sum_{\mathbf{r}'}\omega_n^{(i,j)}(\mathbf{r},\mathbf{r}')\cdot P_{j,\mathbf{r}'},$$

where the weights $\omega_n^{(i,j)}(\mathbf{r},\mathbf{r}')$ denote the average number of recursive calls on a subtree of size j with rank vector $\mathbf{r}'$ when the call is made on a (sub)tree of size n, the rank vector is $\mathbf{r}$ and the root discriminates with respect to coordinate i.

To simplify, for the rest of the section we will assume that the query has no extreme coordinates, that is, $\rho_0 = 0$ (the arguments below can be easily adapted to the case $\rho_0 > 0$, however the details are slightly messier). It is not difficult to show that if we set $P'_{n,\mathbf{r}} = P_{n,\mathbf{r}} - \frac{1}{1-\rho}$, then $P'_{n,\mathbf{r}}$ satisfies the homogeneous recurrence

$$P'_{n,\mathbf{r}} = \frac{1}{nK} \sum_{i=0}^{K-1} \sum_{j=0}^{n-1} \sum_{\mathbf{r}'} \omega_n^{(i,j)}(\mathbf{r},\mathbf{r}') \cdot P'_{j,\mathbf{r}'}, \tag{9}$$

and $P'_{0,\mathbf{r}} = -1/(1-\rho)$, since $P_{0,\mathbf{r}} = 0$. The same can be done for any other toll function $\tau_{n,\mathbf{r}}$ in the recurrence, as long as $\tau_{n,\mathbf{r}} = o(n^\alpha)$. That is, if

$$U_{n,\mathbf{r}} = \tau_{n,\mathbf{r}} + \frac{1}{nK} \sum_{i=0}^{K-1} \sum_{j=0}^{n-1} \sum_{\mathbf{r}'} \omega_n^{(i,j)}(\mathbf{r},\mathbf{r}') \cdot U_{j,\mathbf{r}'},$$

then we can write $U_{n,\mathbf{r}} = U^{[\mathrm{hom}]}_{n,\mathbf{r}} + V_{n,\mathbf{r}}$, with $U^{[\mathrm{hom}]}_{n,\mathbf{r}}$ satisfying the homogeneous recurrence (9) and $V_{n,\mathbf{r}} = \mathcal{O}(\tau_{n,\mathbf{r}}) = o(n^\alpha)$. The initial conditions that $U^{[\mathrm{hom}]}_{n,\mathbf{r}}$ satisfies are of course different from the initial conditions of $U_{n,\mathbf{r}}$.

The computations (most of them given in Appendices A and B) that lead to the closed form for f actually show that

$$f^{[hom]}(z_0,\ldots,z_{t-1}) = c \cdot \left(\prod_{i=0}^{t-1} z_i(1-z_i) \right)^{\alpha/2},$$

with c an arbitrary constant, and $z_i = r_i/n$, give us the general solution of the homogeneous recurrence, subject only to the constraints (a), (b) and (c) given in Proposition 1 since these stem from the symmetries in the weights $\omega_n^{(i,j)}$ of the recurrence (see the discussion in Appendix A). In this sense, the only "arbitrary" initial condition is the one in constraint (d) since it is directly related to the initial values of $U^{[\mathrm{hom}]}_{n,\mathbf{r}}$.

The computations in Appendix C show that $U_{n,\mathbf{r}} = f^{[hom]}(\mathbf{r}/n) \cdot n^\alpha + o(n^\alpha)$ is a solution of the homogeneous recurrence (9), satisfying only the structural constraints (a)–(c) of Proposition 1. To establish that, we only need the closed form for f and the computations given in Appendix C: we do not need to hypothesize the existence of the limit $f = \lim_{n\to\infty} P_{n,\mathbf{r}}/n^\gamma$; the hypothesis is a useful guess to arrive at the closed form for f.

Finally, taking into account constraint (d) gives us the value of c and shows that $P_{n,\mathbf{r}} = f(\mathbf{r}/n) \cdot n^\alpha + o(n^\alpha)$.

We have already discussed at the end of Subsection 3.1 the connection between $P_{n,\mathbf{q}}$ and $P_{n,\mathbf{r}}$. Since $P_{n,\mathbf{r}} = f(r_0/n,\ldots,r_{t-1}/n) \cdot n^\alpha + o(n^\alpha)$, where f is

given in Proposition 2, when the coordinates of the n data points are independent and uniformly distributed we write

$$P_{n,\mathbf{q}} = \sum_{\mathbf{r}} (f(r_0/n, \ldots, r_{t-1}/n) n^{\alpha} (1 + o(1)) \prod_{i=0}^{t-1} \binom{n}{r_i} q_i^{r_i} (1 - q_i)^{n - r_i}$$
$$= n^{\alpha} \nu_{s,t,K} \prod_{i=0}^{t-1} \sum_{0 \le r \le n} \binom{n}{r} q_i^{r} (1 - q_i)^{n-r} \left(\frac{r}{n} \left(1 - \frac{r}{n} \right) \right)^{\alpha/2} + o(n^{\alpha}).$$

Now, the main contribution in each summation of the form

$$\sum_{0 \le r \le n} \binom{n}{r} q_i^{r} (1 - q_i)^{n-r} \left(\frac{r}{n} \left(1 - \frac{r}{n} \right) \right)^{\alpha/2}$$

comes from a neighborhood of size Δ of $\overline{r} = nq_i$. With $\Delta = o(n^{2/3})$ we can apply the de Moivre-Laplace limit theorem [Fel71] to show that the contributions of the ranges $r < \overline{r} - \Delta$ and $r > \overline{r} + \Delta$ are negligible and to compute the contribution from the middle range $\overline{r} - \Delta \le r \le \overline{r} + \Delta$. Finally, we can expand $\frac{r}{n}(1 - \frac{r}{n})$ as $\frac{\overline{r}}{n}(1 - \frac{\overline{r}}{n}) + O(\Delta/n)$ proving thus

$$\sum_{0 \le r \le n} \binom{n}{r} q_i^{r} (1 - q_i)^{n-r} \left(\frac{r}{n} \left(1 - \frac{r}{n} \right) \right)^{\alpha/2} = \left(\frac{\overline{r}}{n} \left(1 - \frac{\overline{r}}{n} \right) \right)^{\alpha/2} (1 + O(\Delta^2/n))$$
$$= (q_i(1 - q_i))^{\alpha/2} (1 + o(1)).$$

The asymptotic estimate for $P_{n,\mathbf{q}}$ given in the last part of Theorem 4 follows

$$P_{n,\mathbf{q}} = \nu_{s,t,K} \cdot \left(\prod_{i=0}^{t-1} q_i (1 - q_i) \right)^{\alpha/2} \cdot n^{\alpha} + o(n^{\alpha}).$$

4 Fixed partial match queries in standard K-d trees

The analysis of the expected cost of a PM search with a fixed query in random standard K-d trees goes along the same lines as the analysis for relaxed K-d trees in the previous section. The major differences stem from the fact that now the query pattern is relevant for the performance of PM searches.

As in Section 3, we will need an intermediate result about the expected cost of random PM queries with t regular coordinates, s_0 extreme coordinates and $K - s = K - (t + s_0)$ unspecified coordinates. The constant factor in the leading term of $\overline{P}_n$ depends on the query pattern $\mathbf{u}(\mathbf{q})$ and thus we shall write $\overline{P}_{n,\mathbf{u}}$ for the expected cost of a random PM query with pattern $\mathbf{u}$ in a random standard K-d tree of size n.

Theorem 6. *Let $\overline{P}_{n,\mathbf{u}} = \mathbb{E}[\mathcal{P}_{,\mathbf{u},\mathbf{Q}}]$ be the expected cost of a PM search in a random standard K-d tree of size n with a random query $\mathbf{Q}$ with pattern $\mathbf{u} = \mathbf{u}(\mathbf{Q})$, in which s coordinates are specified and the remaining $K-s$ coordinates are left unspecified. Of the s specified coordinates exactly t coordinates are independently drawn at random from the same continuous distribution(s) from which data coordinates are drawn, and exactly $s_0 = s - t$ coordinates are extreme (we assume, w.l.o.g., that they are 0). Then*

$$\overline{P}_{n,\mathbf{u}} = \beta_{\mathbf{u}} n^{\alpha(\rho,\rho_0)} + o(n^{\alpha(\rho,\rho_0)}),$$

where the exponent $\alpha(\rho,\rho_0)$ is the unique positive real root in $(0,1)$ of the equation

$$\Big(\alpha(\rho,\rho_0)+2\Big)^{\rho-\rho_0}\Big(\alpha(\rho,\rho_0)+1\Big)^{1-\rho+\rho_0} = 2^{1-\rho_0}, \tag{10}$$

and the $\beta_{\mathbf{u}}$'s are constants that depend on the query pattern $\mathbf{u}$ (and implicitly on s, s_0 and K).

Proof. Given the fixed rule to assign discriminants to nodes in standard K-d trees we need to introduce some notation. We will abbreviate $(i+1) \bmod K$ as $i \oplus 1$. Let $\mathbf{u}^{(i)}$ be the pattern $\mathbf{u}$ shifted to the left i times; in particular we have $\mathbf{u}^{(K)} = \mathbf{u}^{(0)} = \mathbf{u}$. Let $\overline{P}^{(i)}_{n,\mathbf{u}}$ be the expected cost of a PM query with pattern $\mathbf{u}$ in a random standard K-d tree of size n when the root discriminates w.r.t. the i-th coordinate; of course, we are interested in $\overline{P}^{(0)}_{n,\mathbf{u}} = \overline{P}_{n,\mathbf{u}}$.

Let $P^{(i)}_{\mathbf{u}}(z) = \sum_{n\geq 0} \overline{P}^{(i)}_{n,\mathbf{u}} z^n$ be the generating function for the expected costs $\overline{P}^{(i)}_{n,\mathbf{u}}$, $0 \leq i < K$. If $u_i = *$ the search has to continue in both subtrees, if $u_i = E$ the search has to continue in only one subtree (w.l.o.g. we can assume that it is in the left subtree), and if $u_i = S$ the search continues on the left subtree with probability $(j+1)/(n+1)$ and on the right subtree with probability $(n-j)/(n+1)$. This gives us a system of differential equations for the vector $\langle P^{(0)}_{\mathbf{u}}(z), \ldots, P^{(K-1)}_{\mathbf{u}}(z)\rangle$ which can be analyzed using the same techniques as in [FP86] or [CLF89], for instance; the only difference in setting up such a system comes from the extreme coordinates, that is, whenever $u_i = E$. The singularity analysis of this system yields that

$$P^{(i)}_{\mathbf{u}}(z) \sim \eta_{\mathbf{u}}(1-z)^{-(\alpha+1)}, \qquad 0 \leq i < K,$$

around the singularity $z = 1$, where $\alpha = \alpha(\rho,\rho_0)$ is the unique solution in $(0,1)$ of

$$\Big(\alpha(\rho,\rho_0)+2\Big)^{\rho-\rho_0}\Big(\alpha(\rho,\rho_0)+1\Big)^{1-\rho+\rho_0} = 2^{1-\rho_0}.$$

The rest of the proof easily follows from standard singularity analysis techniques [FO90, FS09] with $\beta_{\mathbf{u}} = \eta_{\mathbf{u}}/\Gamma(\alpha+1)$. □

It is not too difficult to prove that $\alpha(\rho,\rho_0)$, with $0 \leq \rho_0 \leq \rho < 1$ is strictly decreasing as ρ_0 increases (for a fixed ρ) and it is also decreasing if ρ increases,

just as in the case of relaxed K-d trees (there, the proof of these facts is straightforward, as a simple closed form for $\alpha(\rho, \rho_0)$ is available).

In particular, $\alpha(0,0) = 1$, $\alpha(\rho,\rho) = 2^{1-\rho} - 1$ and $\lim_{\rho\to 1} \alpha(\rho, g(\rho)) = 0$ for any function g such that $g(x) \le x$. Furthermore, $\alpha(\rho,\rho_0) \ge 2^{1-\rho} - 1$ for any ρ and $\rho_0 \le \rho$.

Since the query pattern is relevant, we can no longer assume that the query $\mathbf{q}$ or its rank vector $\mathbf{r}$ have particularly convenient forms; instead, we will need to introduce ℓ_k for the index of the k-th specified regular coordinate. Thus $q_{\ell_k} \in (0,1)$ for $0 \le k < t$; all other positions correspond to specified extreme coordinates or unspecified coordinates. We can now state the main result of this section.

Theorem 7. *For a query* $\mathbf{q}$ *with rank vector* $\mathbf{r} = (r_0, \ldots, r_{K-1})$ *such that* $r_{\ell_k} = z_{\ell_k} n + o(n)$, $0 < z_{\ell_k} < 1$, *for all* k, $0 \le k < t$, $r_i = o(n)$ *or* $r_i = n - o(n)$ *for exactly* s_0 *coordinates and* $r_i = *$ *for the remaining* $K - s$ *coordinates, the expected cost of the partial match in a random standard* K*-d tree of size* n *is*

$$P_{n,\mathbf{r}} = \nu_{\mathbf{u}(\mathbf{r})} \cdot \left(\prod_{k=0}^{t-1} z_{\ell_k}(1 - z_{\ell_k}) \right)^{\alpha/2} \cdot n^{\alpha} + o(n^{\alpha}),$$

where the exponent $\alpha(\rho,\rho_0)$ *is the same as in Equation* (10) *of Theorem 6, and*

$$\nu_{\mathbf{u}} = \beta_{\mathbf{u}} \frac{\Gamma^t(\alpha+2)}{\Gamma^{2t}(\frac{\alpha}{2}+1)},$$

with $\beta_{\mathbf{u}}$ *the constant factor in the leading term of* $\overline{P}_{n,\mathbf{u}}$ *(see Theorem 6).*

Let $P^{(i)}_{n,\mathbf{r}}$ denote the expected cost of a PM search for a fixed query with rank vector $\mathbf{r}$ in a random standard K-d tree of size n where the root discriminates with respect to coordinate i, $0 \le i < K$. As before the quantity of interest will be $P_{n,\mathbf{r}} := P^{(0)}_{n,\mathbf{r}}$. Reasoning as in Subsection 3.5 and using the same notation as we have used there (e.g., $h(\delta_k)$, $\mathcal{L}^{(i,j)}_{\mathbf{r}}$, ...), we can set up a system of recurrences[3]:

$$P^{(i)}_{n,\mathbf{r}} = 1 + \frac{1}{n}\left[\sum_{j=r_i}^{n-1} \sum_{\mathbf{r}'\in\mathcal{L}^{(i,j)}_{\mathbf{r}}} \prod_{\substack{0\le k<s\\ \ell_k\neq i}} \frac{\binom{j}{r'_{\ell_k}}\binom{n-1-j}{r_{\ell_k}-r'_{\ell_k}}}{\binom{n-1}{r_{\ell_k}}} P^{(i\oplus 1)}_{j,\mathbf{r}'} \right.$$
$$\left. + \sum_{j=0}^{r_i-1} \sum_{\mathbf{r}'\in\mathcal{R}^{(i,j)}_{\mathbf{r}}} \prod_{\substack{0\le k<s\\ \ell_k\neq i}} \frac{\binom{n-1-j}{r'_{\ell_k}}\binom{j+1}{r_{\ell_k}-r'_{\ell_k}}}{\binom{n}{r_{\ell_k}}} P^{(i\oplus 1)}_{n-1-j,\mathbf{r}'} \right], \qquad \text{if } u_i(\mathbf{r}) \neq *,$$

[3] For brevity, we do not make the distinction between regular and extreme coordinates; ℓ_0, ..., ℓ_{s-1} give the indices of specified coordinates.

and

$$P_{n,\mathbf{r}}^{(i)} = 1 + \frac{1}{n}\sum_{j=0}^{n-1}\sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}}\sum_{\boldsymbol{\delta}\in\{0,1,*\}^K}\prod_{0\le k<s} h(\delta_{\ell_k})\frac{\binom{j}{r'_{\ell_k}}\binom{n-1-j}{r_{\ell_k}-\delta_k-r'_{\ell_k}}}{\binom{n-1}{r_{\ell_k}-\delta_k}} \times \left(P_{j,\mathbf{r}'}^{(i\oplus 1)} + P_{n-1-j,\mathbf{r}-\mathbf{r}'-\boldsymbol{\delta}}^{(i\oplus 1)}\right), \qquad \text{if } u_i(\mathbf{r}) = *.$$

Using the same techniques as in the previous section and Appendix A we can show that if the limits

$$f_{\mathbf{u}}^{(i)}(z_{\ell_0},\ldots,z_{\ell_{t-1}}) = \lim_{n\to\infty}\frac{P_{n,\mathbf{r}}^{(i)}}{n^\gamma}$$

exist and are not identically null, with $z_{\ell_k} = \lim_{n\to\infty} r_{\ell_k}/n$, $0 < z_{\ell_k} < 1$, and $\mathbf{u} = \mathbf{u}(\mathbf{r})$, then we get[4]

$$f^{(i)}(z_{\ell_0},\ldots,z_{\ell_{t-1}}) = \begin{cases} \frac{2}{\gamma+1} f^{(i\oplus 1)}(z_{\ell_0},\ldots,z_{\ell_{t-1}}), & \text{if } u_i = *,\\ \frac{1}{\gamma+1} f^{(i\oplus 1)}(z_{\ell_0},\ldots,z_{\ell_{t-1}}), & \text{if } u_i = E,\\ z_i^{\gamma+1}\int_{z_i}^1 f^{(i\oplus 1)}(\ldots,z,\ldots)\frac{dz}{z^{\gamma+2}} & \\ \quad +(1-z_i)^{\gamma+1}\int_0^{z_i} f^{(i\oplus 1)}(\ldots,z,\ldots)\frac{dz}{(1-z)^{\gamma+2}}, & \text{if } u_i = S. \end{cases}$$

The constraints that this system of integral equations must satisfy are analogous to those in the case of relaxed K-d trees. In particular, we must have $\gamma = \alpha(\rho,\rho_0)$, and

(a) For any k, $0 \le k < t$, and $z_{\ell_k} \in (0,1)$, $f(\ldots,z_{\ell_k},\ldots) = f(\ldots,1-z_{\ell_k},\ldots)$.

(b) For any k, $0 \le k < t$,

$$\lim_{z_{\ell_k}\to 0} f(z_{\ell_0},\ldots,z_{\ell_k},\ldots,z_{\ell_{t-1}}) = \lim_{z_{\ell_k}\to 1} f(z_{\ell_0},\ldots,z_{\ell_k},\ldots,z_{\ell_{t-1}}) = 0.$$

(c)

$$\int_0^1\cdots\int_0^1 f_{\mathbf{u}}(y_0,\ldots,y_{t-1})\,dy_0\cdots dy_{t-1} = \beta_{\mathbf{u}}.$$

To solve the system of integral equations and to complete the proof of Theorem 7 we can proceed along similar steps to those in Appendices B and C.

5 Experiments

Our analysis of the expected cost of PM search for fixed queries is only asymptotic and provides the leading order term. We have conducted several experiments in order to investigate for which input sizes we can expect to get relatively

[4]From now on, we shall omit the subscript $\mathbf{u}$ of f to simplify notation.

good predictions from the theoretical analysis. As we will see, the main conclusion of our experiments is that, despite the asymptotic nature of our theoretical results, they do quite a reasonable job at predicting the cost of PM queries, even for inputs of moderate size, e.g., $n \approx 5000$.

Each run of our experiments can be described by a tuple $\langle \mathcal{T}, \mathbf{q}, n, M \rangle$ where $\mathcal{T}$ is a type of K-d trees (standard, relaxed or squarish), $\mathbf{q}$ is the query, n is the size of the trees in the sample and M is the size of the sample. For each run we generate M random K-d trees of type $\mathcal{T}$ and size n. In each tree we perform a PM search with query $\mathbf{q}$, counting the total number of visited nodes and taking the corresponding sample mean $\overline{p}_{n,\mathbf{q}} := 1/M \sum_{i=1}^{M} \mathcal{P}_{n,\mathbf{q}}^{(i)}$, where the $\mathcal{P}_{n,\mathbf{q}}^{(i)}$ are independent identical realizations of $\mathcal{P}_{n,\mathbf{q}}$. Most of the plots that follow depict the normalized mean cost $\overline{c}_{n,\mathbf{q}} := \overline{p}_{n,\mathbf{q}}/n^{\alpha}$, with α the appropriate exponent for each case—that is, depending on $\mathcal{T}$ and the ratios ρ_0 and ρ that correspond to $\mathbf{q}$. For comparison, we depict also the theoretical value $f(q_0, \ldots, q_{t-1}) \sim P_{n,\mathbf{q}}/n^{\alpha}$. In all of these plots one of the most conspicuous features is the dome-like symmetrical shape of the PM search cost along every specified coordinate.

Let us start by reporting results for random relaxed 3-d trees. Figure 2 shows the behavior of the normalized mean cost $\overline{c}_{n,\mathbf{q}}$ for queries with two specified coordinates, x and y; the plot shows the variation of the experimentally measured cost w.r.t. y, as x takes several distinct values ($x = 0, 0.1, 0.25, 0.5$). For $x \neq 0$ we have normalized dividing by $n^{\alpha(2/3,0)} = n^{0.457\cdots}$, whereas for $x = 0$ we have normalized dividing by $n^{\alpha(2/3,1/3)} = n^{0.387\cdots}$. For each value of x we also plot the theoretical prediction $f(x, y)$ (smooth curves). A simple visual inspection of the plots reveals that the experimental data (ragged curves) match the predicted costs quite well. Because of the normalization the curves for $x = 0$ lie somewhere between those for $x = 0.25$ and $x = 0.5$; however, the reader must recall that different normalizations have been used to plot these curves and the mean cost $\overline{p}_{n,\mathbf{q}}$ when $x = 0$ is actually significantly smaller than the mean cost when x is sufficiently away from 0.

Figure 3 plots the relative error in the previous experiment, that is,

$$\epsilon_{n,\mathbf{q}} = \frac{\sqrt{s_{n,\mathbf{q}}^2}}{\overline{p}_{n,\mathbf{q}}},$$

where $s_{n,\mathbf{q}}^2$ is the sample variance. A plausible explanation for the high relative errors that we get in the experiments is that $\mathbb{V}[\mathcal{P}_{n,\mathbf{q}}]$ is likely of order $n^{2\alpha}$ (that has been proven true for 2-d quadtrees [BNS13]) and hence there is no concentration around the expectation. More specifically, as we will later discuss, we conjecture that

$$\mathbb{V}[\mathcal{P}_{n,\mathbf{q}}] = \nu_{s,t,K}^{(2)} \cdot \left(\prod_{i=0}^{t-1} q_i(1 - q_i) \right)^{\alpha/2} \cdot n^{2\alpha} + \text{l.o.t.}$$

for some constant $\nu_{t,s,K}^{(2)}$ and thus $\epsilon_{n,\mathbf{q}} \approx \sqrt{\nu^{(2)}}/\nu$. The experiments give credibility to this hypothesis; moreover, when $x = 0$, the constant ν changes (and

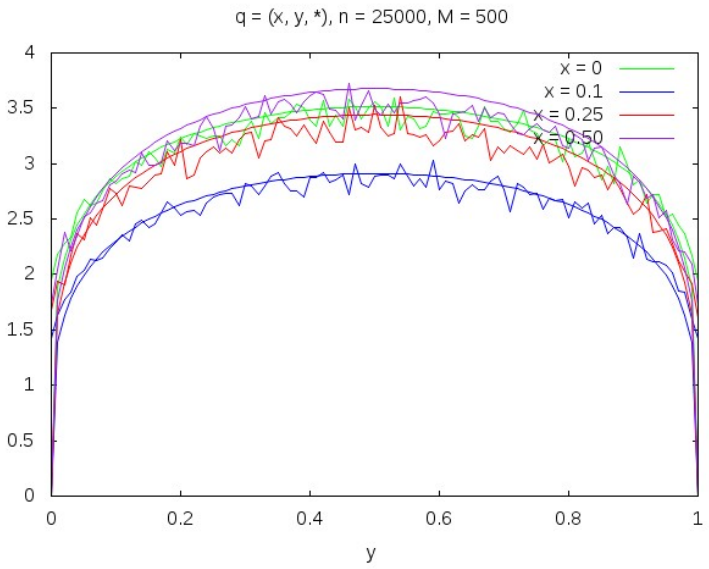

Figure 2: Normalized expected costs (experimental vs. theoretical) for different fixed PM queries in relaxed 3-d trees

likely $\nu^{(2)}$ does too); from the experiment it seems that their ratio would be larger than the ratio $\sqrt{\nu^{(2)}_{s,s,K}}/\nu_{s,s,K}$.

Figure 4 shows how the experimental results get closer to the theoretical predicted expected cost for increasing values of n. As we already have mentioned for moderate input sizes such as $n = 5000$, the experimental mean costs are already quite close to the theoretical expectation (which is approximated by its leading order term). The queries used in this experiment are of the form $\mathbf{q} = (0.25, y, *)$.

Figures 5 and 6 are similar to Figures 2 and 3, but for standard 3-d trees. In contrast with relaxed K-d trees, the pattern of regular, extreme and unspecified coordinates in the query is relevant. In Figure 5 we try to capture that dependency by considering three different queries which differ in their pattern: $\mathbf{q} = (0.25, y, *)$, $\mathbf{q} = (0.25, *, y)$ and $\mathbf{q} = (*, 0.25, y)$. In all three runs the normalizing factor of the mean costs is $n^{-\alpha(2/3,1/3)} = n^{-0.3146\cdots}$. We can appreciate again that we obtain qualitatively similar behavior of the average costs as the one for relaxed K-d trees, just as our theoretical results predict. In this case it is important to remark that both the average cost and the variance are smaller than the ones for relaxed K-d trees (the exponent $\alpha(2/3, 0) = 0.395\ldots$ for standard 3-d trees, whereas $\alpha(2/3, 0) = 0.457\ldots$ for relaxed 3-d trees). Furthermore, the constant coefficient ν in $P_{n,\mathbf{q}}$ depends on the pattern $\mathbf{u}(\mathbf{q})$ (because ν depends on β), a dependency that is well reflected in the experi-

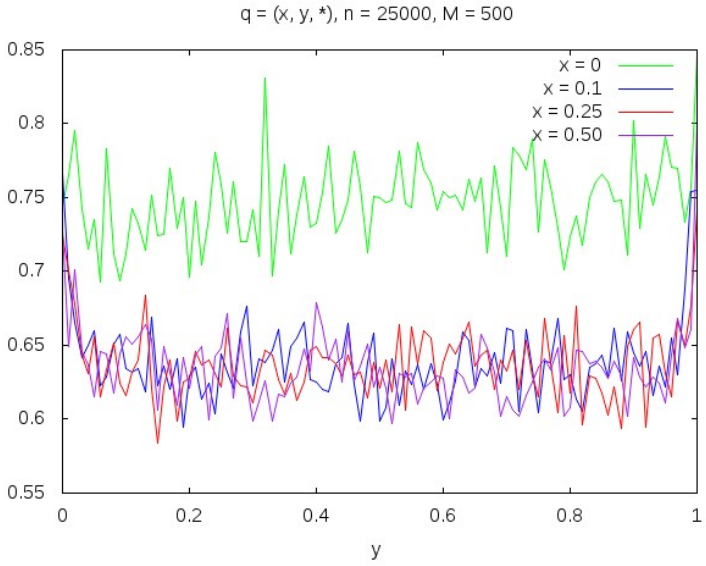

Figure 3: Relative errors of measured costs for different fixed PM queries in relaxed 3-d trees

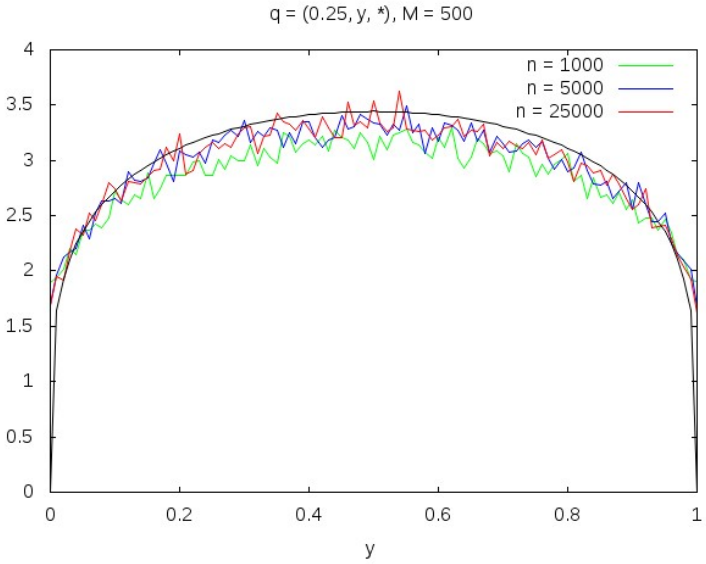

Figure 4: Normalized expected cost (experimental vs. theoretical) for fixed PM queries in relaxed 3-d trees of different sizes.

mental data and well predicted by the theoretical asymptotic expected costs. Like in relaxed K-d trees, a plausible (and analogous) hypothesis for $\mathbb{V}[\mathcal{P}_{n,\mathbf{q}}]$ in standard K-d trees leads to $\epsilon_n \to c$ for some constant c not depending on the actual values of the specified regular coordinates of $\mathbf{q}$, but only on its pattern. Again the experiments support this hypothesis. A remark is in order here: while we have refrained from giving further details about $\beta_{\mathbf{u}}$ (and thus about $\nu_{\mathbf{u}}$), the values β_{SS*}, β_{S*S} and β_{*SS} are readily available from the literature (and related because the three patterns are circular shifts one from the other [CH06]). From these we can obtain the theoretical curves against which to compare the experimental data.

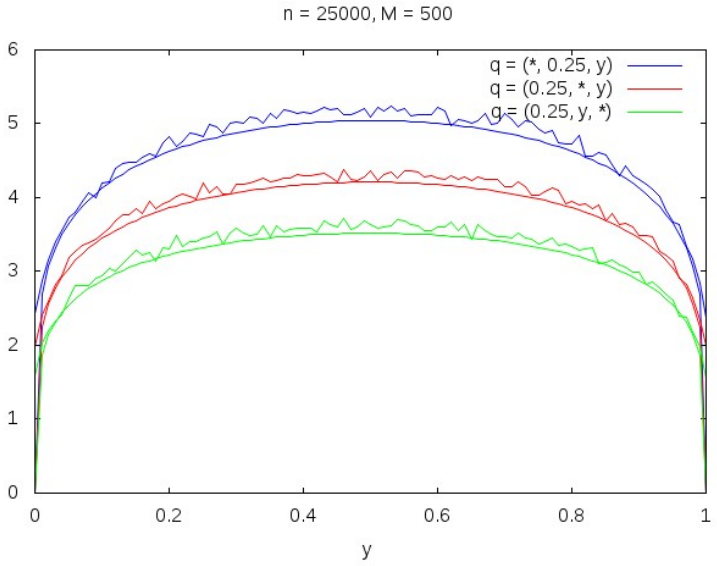

Figure 5: Normalized expected cost (experimental vs. theoretical) for fixed PM queries in standard 3-d trees with different patterns.

The last variant of K-d trees that we have considered in our experiments is squarish K-d trees. Here we have no theoretical results to compare with, but experiments can be used to test several different hypotheses. Recall that the expected cost of a random PM query is $\Theta(n^{1-s/K})$, and experiments could give useful hints about the expected cost of fixed PM queries in these K-d trees. The experiments for squarish K-d trees hide a "surprise": $P_{n,\mathbf{q}}$ seems not to depend on $\mathbf{q}$ at all, only on s and K. That is in sharp contrast with what can be analytically proved and experimentally observed for relaxed and standard K-d trees. Figure 7 shows the normalized mean cost for squarish 3-d trees; the normalizing factor is $n^{-1/3}$. A combinatorial argument shows that the order of specified and unspecified coordinates in the query is irrelevant, like in relaxed K-d trees and thus we anticipated that $P_{n,\mathbf{q}}$ would not depend on $\mathbf{u}(\mathbf{q})$ for

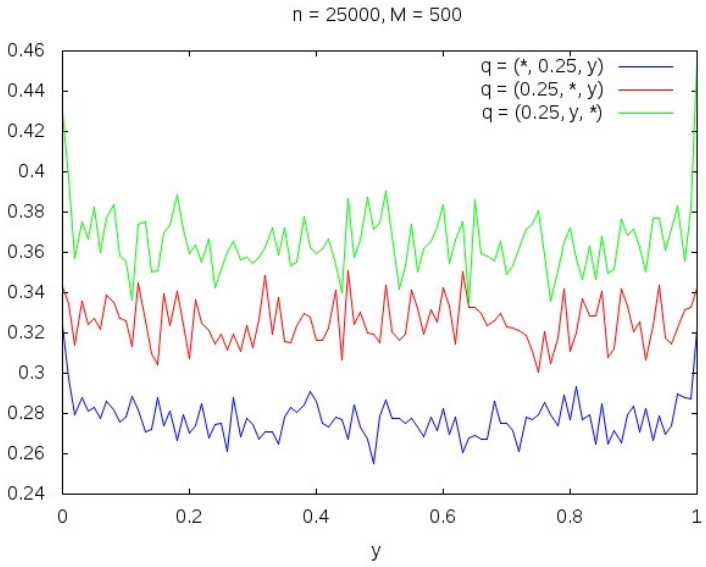

Figure 6: Relative errors of the expected costs for fixed PM queries in standard 3-d trees with different patterns.

squarish K-d trees. The (initially) surprising fact is that $P_{n,\mathbf{q}}$ seems not to depend on the values of the specified regular coordinates either—at least, the experiments indicate so. This drastically different behavior is not so surprising in the light that squarish K-d trees are far more balanced than the other variants (the exponent $\alpha = 1 - s/K$ is optimal for the expected cost of a PM query with s regular coordinates in any multidimensional data structure). The experiments support then the conjecture that

$$P_{n,\mathbf{q}} \sim \nu_{t,s,K} n^{1-s/K}$$

in squarish K-d trees. The exponent in n is always $1 - s/K$, irrespective of s_0 and t, and the constant ν does depend on s and t (or $s_0 = s - t$), but not on the actual values of the regular coordinates of $\mathbf{q}$. In fact, we conjecture that this is also the case with any other multidimensional data structure that attains the optimal exponent $\alpha = 1 - s/K$ for PM search.

Besides $K = 3$, we have also conducted experiments in relaxed, standard and squarish K-d trees for higher dimensional settings, all showing qualitatively identical results to those reported above.

All the programs used in the experiments were written in the `C++` programming language and compiled with the `GNU gcc` compiler version `v4.4.3`. The experiments were run on a Pentium Genuine Intel `x86_64` 64-bit dual 32K core processor.

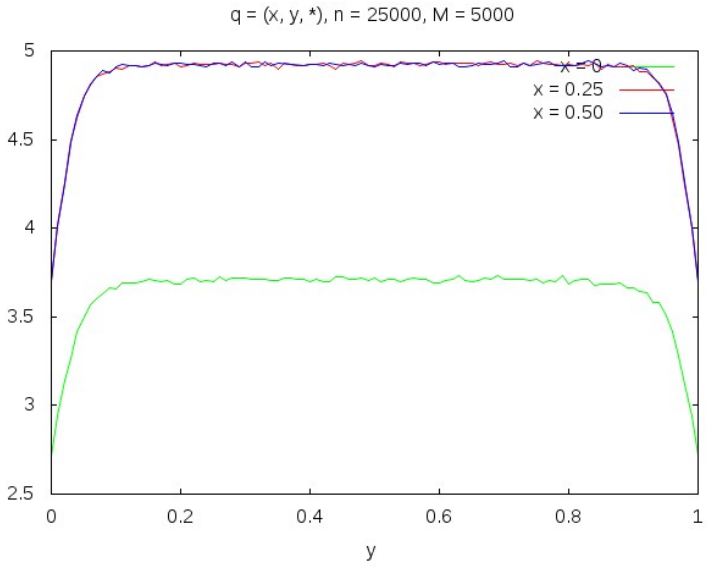

Figure 7: The average cost of fixed PM queries in squarish 3-d trees

6 Conclusions

For a period of 25 years, the probabilistic analysis of partial match and other associative queries in multidimensional data structures focused on random queries, a program which has been extremely successful and has yielded a good share of interesting results. In the last four years several authors [CJ11, BNS13, DJM14] have tackled the analysis of partial match with fixed queries; this paper is one additional contribution in that direction. The three mentioned papers, together with the present one, have uncovered the pervasiveness of the function

$$\psi(z) = (z(1-z))^{\alpha/2}$$

at the heart of the cost of PM searches with fixed queries for several different data structures, namely, 2-d quadtrees, relaxed K-d trees and standard K-d trees. The function appears in the expected cost $P_{n,\mathbf{q}}$ (in all three mentioned variants); Broutin *et al.* [BNS13] have also proved that it is involved in a very deep sense in the case of 2-d quadtrees (an analogous result holds for standard 2-d trees): there exists a random variable Φ such that, for a query $\mathbf{q} = (z, *)$ (or $\mathbf{q} = (*, z)$),

$$\left(\frac{\mathcal{P}_{n,\mathbf{q}}}{\nu \cdot n^{\alpha}}, z \in (0,1) \right) \xrightarrow{(d)} (\psi(z) \cdot \Phi, z \in (0,1))$$

where the exponent $\alpha = \alpha(1/2) = (\sqrt{17} - 3)/2$ is the one corresponding to a random PM query in a random 2-d quadtree and ν is a constant (we don't give

its explicit form here, but it has been computed in [BNS13]). By $X_n \xrightarrow{(d)} X$ we mean that the sequence of random variables X_n converges in distribution to the random variable X, as $n \to \infty$. The same notation is used for the convergence of a stochastic process $(X_n(z), z \in (0,1))$ to a continuous random function X.

We conjecture that this is the situation for all hierarchical multidimensional data structures (quadtrees and many variants of K-d trees, for instance) such that $\alpha(x) > 1 - x$. In fact, we conjecture that

$$\left(\frac{\mathcal{P}_{n,\mathbf{q}}}{n^{\alpha}}\right) \xrightarrow{(d)} \frac{\Gamma^s(\alpha+2)}{\Gamma^{2s}(\alpha/2+1)} \left(\prod_{i:q_i \neq *, q_i \notin \{0,1\}} \psi(q_i)\right) \cdot \Phi_{\mathbf{u}(\mathbf{q})}, \tag{11}$$

where $\Phi_{\mathbf{u}(\mathbf{q})}$ is the random variable such that

$$\frac{\overline{\mathcal{P}}_{n,\mathbf{q}}}{n^{\alpha(\rho,\rho_0)}} \xrightarrow{(d)} \Phi_{\mathbf{u}(\mathbf{q})},$$

and $\overline{\mathcal{P}}_n$ denotes the cost of a random PM query with pattern $\mathbf{u}$ (in some data structures such as quadtrees and relaxed K-d trees the query pattern is irrelevant, but this formulation is the most general). It is also reasonable to conjecture that for expected costs,

$$\left(\frac{P_{n,\mathbf{q}}}{n^{\alpha}}\right) \to \beta_{\mathbf{u}(\mathbf{q})} \cdot \frac{\Gamma^s(\alpha+2)}{\Gamma^{2s}(\alpha/2+1)} \left(\prod_{i:q_i \neq *, q_i \notin \{0,1\}} \psi(q_i)\right), \qquad n \to \infty$$

where $\beta_{\mathbf{u}}$ and α come from the expected cost of a random PM query, namely,

$$\overline{P}_{n,\mathbf{u}} = \mathbb{E}\big[\overline{\mathcal{P}}_{n,\mathbf{u}}\big] = \beta_{\mathbf{u}} n^{\alpha} + \text{l.o.t.}$$

Indeed, we have proved in this paper that such is the case for relaxed and standard K-d trees.

We believe (see Section 5) that the conjecture (11) above does not hold in squarish K-d trees and other multidimensional data structures such that $\alpha(x) = 1 - x$. Instead, we conjecture that, in these cases,

$$\mathcal{P}_{n,\mathbf{q}}/n^{1-s/K} \xrightarrow{(d)} c_{\mathbf{u}(\mathbf{q})} \tag{12}$$

for some constant $c_{\mathbf{u}}$ that only depends on ρ_0 and ρ for squarish K-d trees and, more generally, might depend on the pattern $\mathbf{u}$ of the query for other data structures. Also, if we consider the expected cost of fixed PM queries in such data structures, it is natural to anticipate that $P_{n,\mathbf{q}} = c_{\mathbf{u}} n^{1-s/K} + \text{l.o.t.}$.

The two conjectures discussed above (Eqs. (11) and (12)) are far-reaching as they would apply to whole families of multidimensional data structures, and the analysis should be conducted at a very high level of abstraction. We are currently working on the distributional analysis of PM queries in standard and relaxed K-d trees, with the goal of proving the conjecture for these particular

variants of K-d trees. Somewhat simpler than the distributional analysis, the analysis of the variance and other higher order moments of $\mathcal{P}_{n,\mathbf{q}}$ can be carried out using the same techniques as in this paper, without the need for more sophisticated tools such as the contraction method used in [BNS13]. Another family of K-d trees for which we already have encouraging preliminary results is K-d-t trees [CLF89] (they result from the addition of a local rebalancing rule of small subtrees to a regular variant of K-d trees, e.g., relaxed K-d-t trees). We also plan to analyze PM searches with fixed queries in other multidimensional data structures in the near future.

Another interesting line of research is to extend the results about partial match queries to other associative queries with fixed parameters, e.g., nearest neighbor queries for a given query point $\mathbf{q}$ or orthogonal range queries with a fixed hyperrectangle query $Q = [\ell_0, u_0] \times \cdots \times [\ell_{K-1}, u_{K-1}]$.

Acknowledgments

We are very thankful to the two anonymous reviewers of this manuscript for their detailed reports and useful suggestions.

References

[Ben75] J. L. Bentley. Multidimensional binary search trees used for associative retrieval. *Communications of the ACM*, 18(9):509–517, 1975.

[BF74] J.L. Bentley and R.A. Finkel. Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

[BNS13] N. Broutin, R. Neininger, and H. Sulzbach. A limit process for partial match queries in random quadtrees and 2-d trees. *Annals of Applied Probability*, 23(6):2560–2603, 2013.

[CDZC01] P. Chanzy, L. Devroye, and C. Zamora-Cura. Analysis of range search for random k-d trees. *Acta Informatica*, 37(4–5):355–383, 2001.

[CH06] H.-H. Chern and H.-K. Hwang. Partial match queries in random k-d trees. *SIAM Journal on Computing*, 35(6):1440–1466, 2006.

[CJ11] N. Curien and A. Joseph. Partial match queries in two-dimensional quadtrees: A probabilistic approach. *Advances in Applied Probability*, 43(1):178–194, 2011.

[CLF89] W. Cunto, G. Lau, and Ph. Flajolet. Analysis of kdt-trees: kd-trees improved by local reorganisations. In F. Dehne, J.-R. Sack, and N. Santoro, editors, *Workshop on Algorithms and Data Structures (WADS'89)*, volume 382 of *Lecture Notes in Computer Science*, pages 24–38. Springer-Verlag, 1989.

[DECM98] A. Duch, V. Estivill-Castro, and C. Martínez. Randomized k-dimensional binary search trees. In K.-Y. Chwa and O.H. Ibarra, editors, *Proceedings of the 9$^{\text{th}}$ International Symposium on Algorithms and Computation (ISAAC)*, volume 1533 of *Lecture Notes in Computer Science*, pages 199–208. Springer-Verlag, 1998.

[DJM14] A. Duch, R. M. Jiménez, and C. Martínez. Selection by rank in k-dimensional binary search trees. *Random Structures & Algorithms*, 45(1):14–37, 2014.

[DJZC00] L. Devroye, J. Jabbour, and C. Zamora-Cura. Squarish k-d trees. *SIAM Journal on Computing*, 30(5):1678–1700, 2000.

[DLM14] A. Duch, G. Lau, and C. Martínez. On the average performance of fixed partial match queries in random relaxed k-d trees. In M. Bousquet-Mélou and M. Soria, editors, *Proceedings of the 25$^{\text{rd}}$ International Meeting on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA)*, Discrete Mathematics & Theoretical Computer Science (Proceedings), pages 103–114, 2014.

[DM02] A. Duch and C. Martínez. On the average performance of orthogonal range search in multidimensional data structures. *Journal of Algorithms*, 44(1):226–245, 2002.

[Fel71] W. Feller. *An Introduction to Probability Theory and its Applications*. John Wiley & Sons, New York, NY, 1971.

[FO90] Ph. Flajolet and A. Odlyzko. Singularity analysis of generating functions. *SIAM Journal on Discrete Mathematics*, 3(1):216–240, 1990.

[FP86] Ph. Flajolet and C. Puech. Partial match retrieval of multidimensional data. *Journal of the ACM*, 33(2):371–407, 1986.

[FS09] Ph. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[JKK92] N. L. Johnson, S. Kotz, and A.W. Kemp. *Univariate Discrete Distributions*. John Wiley & Sons, New York, NY, 2nd edition, 1992.

[MPP01] C. Martínez, A. Panholzer, and H. Prodinger. Partial match queries in relaxed multidimensional search trees. *Algorithmica*, 29(1–2):181–204, 2001.

A Getting the integral equation (8)

The hypothesis of Proposition 1 is that there exists some γ such that

$$\lim_{n\to\infty} n^{-\gamma} P_{n,\mathbf{r}} = f(z_0, \ldots, z_{t-1})$$

exists and is not identically null, with $z_i = \lim_{n\to\infty} r_i/n \in (0,1)$, $0 \le i < t$. We shall also assume here that all $r_i \le n/2$, for otherwise we can replace r_i by $n - r_i$.

First of all, in the asymptotic regime, when $r_i = o(n)$, the probability that we recursively continue the PM search in the right subtree is $o(1)$, so we can assume all extremal ranks are $r_i = 0$ $(t \le i < s)$, and thus we can rewrite the recurrence as

$$\begin{aligned} P_{n,\mathbf{r}} \sim 1 + \frac{1}{nK}\Bigg[& \sum_{0\le i<t} \Big(\sum_{j=r_i}^{n-1} \sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}} \pi_L^{(i,j)}(\mathbf{r},\mathbf{r}') P_{j,\mathbf{r}'} \\ & \qquad + \sum_{j=0}^{r_i-1} \sum_{\mathbf{r}'\in\mathcal{R}_{\mathbf{r}}^{(i,j)}} \pi_R^{(i,j)}(\mathbf{r},\mathbf{r}') P_{n-1-j,\mathbf{r}'} \Big) \\ & + \sum_{t\le i<s} \sum_{j=0}^{n-1} \sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}} \pi_L^{(i,j)}(\mathbf{r},\mathbf{r}') P_{j,\mathbf{r}'} \\ & + \sum_{s\le i<K} \sum_{j=0}^{n-1} \sum_{\langle \mathbf{r}',\mathbf{r}''\rangle\in\mathcal{B}_{\mathbf{r}}^{(i,j)}} \pi_B^{(i,j)}(\mathbf{r},\mathbf{r}',\mathbf{r}'') \left(P_{j,\mathbf{r}'} + P_{n-1-j,\mathbf{r}''}\right) \Bigg]. \end{aligned}$$

A second simplification comes from the realization that $\pi_L^{(i,j)}(\mathbf{r},\mathbf{r}')$, $\pi_R^{(i,j)}(\mathbf{r},\mathbf{r}')$ are highly concentrated around the expected value of $\mathbf{r}'$; in particular,

$$\sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}} \pi_L^{(i,j)}(\mathbf{r},\mathbf{r}') P_{j,\mathbf{r}'} \sim P_{j,\overleftarrow{\mathbf{r}}}$$

where $\overleftarrow{r_i} = r_i$ and $\overleftarrow{r_k} = \frac{j}{n} r_k$ for $k \ne i$.

Similarly,

$$\sum_{\mathbf{r}'\in\mathcal{R}_{\mathbf{r}}^{(i,j)}} \pi_R^{(i,j)}(\mathbf{r},\mathbf{r}') P_{n-1-j,\mathbf{r}'} \sim P_{n-1-j,\overrightarrow{\mathbf{r}}},$$

where $\overrightarrow{r_i} = r_i - j - 1$ and $\overrightarrow{r_k} = \frac{n-1-j}{n} r_k$ for $k \ne i$. Last but not least,

$$\begin{aligned} \sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}} \sum_{\boldsymbol{\delta}\in\{0,1\}^s} \prod_{0\le k<s} h(\delta_k) \frac{\binom{j}{r'_k}\binom{n-1-j}{r_k-\delta_k-r'_k}}{\binom{n-1}{r_k-\delta_k}} \left(P_{j,\mathbf{r}'} + P_{n-1-j,\mathbf{r}-\mathbf{r}'-\boldsymbol{\delta}}\right) \\ \sim P_{j,\overleftrightarrow{\mathbf{r}}} + P_{n-1-j,\mathbf{r}-\overleftrightarrow{\mathbf{r}}}, \end{aligned}$$

where $\overleftrightarrow{r_k} = \frac{j}{n} r_k$, $0 \le k < s$. Setting $C_{n,\mathbf{r}} := 0$ if $n = 0$, and $C_{n,\mathbf{r}} := n^{-\gamma} P_{n,\mathbf{r}}$ if $n > 0$,

$$C_{n,\mathbf{r}} \sim \frac{1}{n^\gamma} + \frac{1}{nK}\left[\sum_{0\le i<t}\left(\sum_{j=r_i}^{n-1} C_{j,\overleftarrow{\mathbf{r}}}\cdot\left(\frac{j}{n}\right)^\gamma + \sum_{j=0}^{r_i-1} C_{n-1-j,\overrightarrow{\mathbf{r}}}\cdot\left(\frac{n-1-j}{n}\right)^\gamma\right)\right] \tag{13}$$

$$+ \frac{s_0}{nK}\cdot\sum_{j=0}^{n-1} C_{j,\overleftarrow{\mathbf{r}}}\cdot\left(\frac{j}{n}\right)^\gamma$$

$$+ \frac{(K-s)}{nK}\cdot\sum_{j=0}^{n-1}\left(C_{j,\overleftrightarrow{\mathbf{r}}}\cdot\left(\frac{j}{n}\right)^\gamma + C_{n-1-j,\mathbf{r}-\overleftrightarrow{\mathbf{r}}}\cdot\left(\frac{n-1-j}{n}\right)^\gamma\right).$$

Notice that when $t \le i < s$, we assume $r_i = 0$ and thus $\overleftarrow{r_i} = 0$ as well. Now, since $t > 0$,

$$\lim_{n\to\infty}\frac{P_{n,\mathbf{r}}}{n^\gamma} = f(z_0,\ldots,z_{t-1})$$

exists and it is not identically null, by hypothesis. If we substitute $C_{n,\mathbf{r}}$ by $f\left(\frac{r_0}{n},\ldots,\frac{r_{t-1}}{n}\right)$ then

$$f\left(\frac{r_0}{n},\ldots,\frac{r_{t-1}}{n}\right) \sim \frac{1}{n^\gamma}$$

$$+ \frac{1}{nK}\left[\sum_{0\le i<t}\left(\sum_{j=r_i}^{n-1} f\left(\frac{r_0}{n},\ldots,\frac{r_i}{j},\ldots,\frac{r_{t-1}}{n}\right)\cdot\left(\frac{j}{n}\right)^\gamma\right.\right.$$

$$\left.\left. + \sum_{j=0}^{r_i-1} f\left(\frac{r_0}{n},\ldots,\frac{r_i-j-1}{n-1-j},\ldots,\frac{r_{t-1}}{n}\right)\cdot\left(\frac{n-1-j}{n}\right)^\gamma\right)\right]$$

$$+ \frac{s_0}{nK}\cdot\sum_{j=0}^{n-1} f\left(\frac{r_0}{n},\ldots,\frac{r_{t-1}}{n}\right)\cdot\left(\frac{j}{n}\right)^\gamma$$

$$+ \frac{(K-s)}{nK}\cdot\sum_{j=0}^{n-1}\left(f\left(\frac{r_0}{n},\ldots,\frac{r_{t-1}}{n}\right)\cdot\left(\frac{j}{n}\right)^\gamma\right.$$

$$\left. + f\left(\frac{r_0}{n},\ldots,\frac{r_{t-1}}{n}\right)\cdot\left(\frac{n-1-j}{n}\right)^\gamma\right).$$

Passing to the limit when $n \to \infty$, with $z_i = \lim_{n\to\infty}(r_i/n)$, we replace sums

by integrals and thus

$$
\begin{aligned}
f\left(z_0,\ldots,z_{t-1}\right) = \frac{1}{K}\Bigg[& \sum_{0\le i<t}\Big(\int_{z_i}^{1} f\left(z_0,\ldots,\frac{z_i}{z},\ldots,z_{t-1}\right)\cdot z^{\gamma}\,dz \\
& \qquad + \int_{0}^{z_i} f\left(z_0,\ldots,\frac{z_i-z}{1-z},\ldots,z_{t-1}\right)\cdot(1-z)^{\gamma}\;dz\Big) \\
& + s_0\cdot\int_0^1 f\left(z_0,\ldots,z_{t-1}\right)\cdot z^{\gamma}\,dz \\
& + (K-s)\cdot\int_0^1\Big(f\left(z_0,\ldots,z_{t-1}\right)\cdot z^{\gamma} \\
& \qquad + f\left(z_0,\ldots,z_{t-1}\right)\cdot(1-z)^{\gamma}\Big)\,dz\Bigg],
\end{aligned}
$$

which can be further manipulated to give

$$
\begin{aligned}
f\left(z_0,\ldots,z_{t-1}\right) = \frac{1}{K}\Bigg[& \sum_{0\le i<t}\Big(\int_{z_i}^{1} f\left(z_0,\ldots,\frac{z_i}{z},\ldots,z_{t-1}\right)\cdot z^{\gamma}\,dz \\
& \qquad + \int_{0}^{z_i} f\left(z_0,\ldots,\frac{z_i-z}{1-z},\ldots,z_{t-1}\right)\cdot(1-z)^{\gamma}\;dz\Big) \\
& + s_0\cdot f\left(z_0,\ldots,z_{t-1}\right)\cdot\frac{1}{\gamma+1} \\
& + (K-s)\cdot f\left(z_0,\ldots,z_{t-1}\right)\cdot\frac{2}{\gamma+1}\Bigg].
\end{aligned}
$$

Hence

$$
\begin{aligned}
f(z_0,\ldots,z_{t-1}) = \lambda\Bigg[& \sum_{0\le i<t}\Big(\int_{z_i}^{1} f\left(z_0,\ldots,\frac{z_i}{z},\ldots,z_{t-1}\right)\cdot z^{\gamma}\,dz \\
& \qquad + \int_{0}^{z_i} f\left(z_0,\ldots,\frac{z_i-z}{1-z},\ldots,z_{t-1}\right)\cdot(1-z)^{\gamma}\;dz\Big)\Bigg],
\end{aligned}
$$

with

$$
\lambda = \frac{1}{K}\frac{1}{1-\frac{2(K-s)+s_0}{K(\gamma+1)}}.
$$

Furthermore,

$$
\begin{aligned}
f(z_0,\ldots,z_{t-1}) = \lambda\sum_{i=0}^{t-1}\Bigg\{ & z_i^{\gamma+1}\int_{z_i}^{1} f(z_0,\ldots,z_{i-1},z,z_{i+1},\ldots,z_{t-1})\frac{dz}{z^{\gamma+2}} \\
& + (1-z_i)^{\gamma+1}\int_0^{z_i} f(z_0,\ldots,z_{i-1},z,z_{i+1},\ldots,z_{t-1})\frac{dz}{(1-z)^{\gamma+2}}\Bigg\}.
\end{aligned}
$$

with the substitution $z := z_i/z$ in the first integral and $z := (z_i - z)/(1 - z)$ in the second.

Besides the integral equation (8), the properties of $P_{n,\mathbf{r}}$ translate into several constraints that $f(z_0, \ldots, z_{t-1})$ must satisfy:

(a) The function f is symmetric with respect to any permutation of its arguments.

(b) For any i, $0 \le i < t$, and $z_i \in (0, 1)$, $f(z_0, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_{t-1}) = f(z_0, \ldots, z_{i-1}, 1 - z_i, z_{i+1}, \ldots, z_{t-1})$.

(c) For any i, $0 \le i < t$,

$$\lim_{z_i \to 0} f(z_0, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_{t-1}) = \lim_{z_i \to 1} f(z_0, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_{t-1}) = 0.$$

(d)
$$\int_0^1 \cdots \int_0^1 f(y_0, \ldots, y_{t-1})\, dy_0 \cdots dy_{t-1} = \beta(\rho, \rho_0).$$

Constraint (d) follows because of (6). In fact, we must have $\gamma = \alpha(\rho, \rho_0)$, for otherwise we would have a contradiction with our hypothesis: either we have that $\lim_{n\to\infty} P_{n,\mathbf{r}}/n^\gamma = 0$ or that limit does not exist ($\to \infty$). Also, because $\gamma = \alpha(\rho, \rho_0) =: \alpha$ we must have,

$$\lambda = \frac{\alpha + 2}{2t},$$

since

$$1 - \frac{2(K - s) + s_0}{K(\gamma + 1)} = \frac{2t}{K(\alpha + 2)}.$$

Constraint (c) follows from inductive reasoning. Suppose that for any rank vector $\mathbf{r}'$ with $s_0 + 1$ extreme values we have $P_{n,\mathbf{r}'} = \Theta\left(n^{\alpha(\rho,\rho_0+1/K)}\right)$. Since setting $z_i = 0$ or $z_i = 1$ corresponds to one more extreme rank, dividing $P_{n,\mathbf{r}}$ by $n^{-\alpha(\rho,\rho_0)}$ yields that f is 0, because $\alpha(\rho, \rho_0 + 1/K) < \alpha(\rho, \rho_0)$. To prove the basis of this induction, we must analyze the case when all the specified coordinates of a query are extreme. The recurrence for $P_{n,\mathbf{r}}$ in this case ($s_0 = s$) is greatly simplified. Indeed, for such queries we have

$$P_{n,\mathbf{r}} = 1 + \frac{s}{nK} \sum_{j=0}^{n-1} P_{j,\mathbf{r}} + \frac{K - s}{nK} \sum_{j=0}^{n-1} (P_{j,\mathbf{r}} + P_{n-1-j,\mathbf{r}})$$

as the query (actually, its rank vector) does not change as we proceed recursively with the PM search; moreover, whenever the discriminant at the root is one of the specified extreme coordinates we will systematically continue in the left subtree. The solution of the recurrence above is straightforward:

$$P_{n,\mathbf{r}} = \Theta(n^{1-\rho_0}),$$

that is, $P_{n,\mathbf{r}} = \Theta\left(n^{\alpha(\rho_0,\rho_0)}\right)$, as we wanted to show. It is also interesting to note that constraint (c) can also be proved as a consequence of the symmetries (a) and (b), and the symmetries of the weights of the recurrence lead to constraints (a) and (b).

B Solving the integral equation (8)

In order to solve the integral equation (8) given in Proposition 1, together with constraints (a)–(d) we transform it into an equivalent partial differential equation (PDE).

For any function $f(z_0, z_1, \ldots, z_{t-1})$ let

$$L_i[f] := z_i^{\alpha+1} \int_{z_i}^{1} f(z_0, \ldots, z_{i-1}, z, z_{i+1}, \ldots, z_{t-1}) \, \frac{dz}{z^{\alpha+2}},$$

and, similarly let

$$R_i[f] := (1 - z_i)^{\alpha+1} \int_{0}^{z_i} f(z_0, \ldots, z_{i-1}, z, z_{i+1}, \ldots, z_{t-1}) \, \frac{dz}{(1-z)^{\alpha+2}}.$$

If we set $T := \lambda \sum_{i=0}^{t-1} (L_i + R_i)$ where $\lambda = \frac{\alpha+2}{2t}$ then the function f we are looking for is a non-trivial solution to the fix-point equation $f = T[f]$ with the constraints (a)–(d).

Let us now assume that the solution to the integral equation is a function in separable variables, namely $f(z_0, z_1, \ldots, z_{t-1}) = \phi_0(z_0) \cdot \phi_1(z_1) \cdots \phi_{t-1}(z_{t-1})$. Because of the symmetry of f (constraint (a)), it follows that we can safely assume $\phi_0 = \phi_1 = \cdots = \phi_{t-1} =: \phi$. Furthermore, because of constraint (b), we must have $\phi(z) = \phi(1-z)$ for any $z \in (0, 1)$. We must also have $\lim_{z\to 0} \phi(z) = 0$ to satisfy constraint (c).

Going back to the integral equation, if we denote $\phi_i := \phi(z_i)$ we must have

$$\phi_0 \cdot \phi_1 \cdots \phi_{t-1} = \lambda \sum_{i=0}^{t-1} \phi_0 \cdots \phi_{i-1} \cdot \phi_{i+1} \cdots \phi_{t-1} \Big(L_i[\phi_i] + R_i[\phi_i] \Big).$$

If, for all i, $0 \le i < t$,

$$\phi_i = t\lambda \Big(L_i[\phi_i] + R_i[\phi_i] \Big), \tag{14}$$

then

$$\lambda \sum_{i=0}^{t-1} \phi_0 \cdots \phi_{i-1} \cdot \phi_{i+1} \cdots \phi_{t-1} \cdot \Big(L_i[\phi_i] + R_i[\phi_i] \Big) = \lambda \sum_{i=0}^{t-1} \frac{\phi_0 \cdots \phi_{t-1}}{t\lambda} = \phi_0 \cdots \phi_{t-1}.$$

The solution of (14), namely, the solution of

$$\phi(z) = t\lambda \left(z^{\alpha+1} \int_{z}^{1} \phi(u) \frac{du}{u^{\alpha+2}} + (1-z)^{\alpha+1} \int_{0}^{z} \phi(u) \frac{du}{(1-u)^{\alpha+2}} \right)$$

can be obtained by solving the equivalent ordinary differential equation that we obtain applying the operator

$$\Phi_i[g(z_i)] := z_i(1-z_i)\frac{d^2g}{dz_i^2} + \alpha(2z_i-1)\frac{dg}{dz_i} - \alpha(\alpha+1)g(z_i),$$

to both sides. The linear operator Φ allows us to remove the integrals in L_i and R_i:

$$\begin{aligned}\Phi_i[L_i[g]] &= (z_i-1)\frac{dg}{dz_i} - \alpha g\\ \Phi_i[R_i[g]] &= z_i\frac{dg}{dz_i} - \alpha g\\ \Phi_i[(L_i+R_i)[g]] &= (2z_i-1)\frac{dg}{dz_i} - 2\alpha g.\end{aligned}$$

In particular, we obtain the following ODE for $\phi(z)$, after rearranging:

$$z(1-z)\phi''(z) + \alpha(2z-1)\phi'(z) - \alpha(\alpha+1)\phi(z) = t\lambda\Big((2z-1)\phi'(z) - 2\alpha\phi(z)\Big),$$

or more conveniently,

$$z(1-z)\phi''(z) + (\alpha - t\lambda)(2z-1)\phi'(z) - \alpha(\alpha+1-2t\lambda)\phi(z) = 0,$$

with the initial condition $\phi(0) = 0$. Again we have a second order linear hypergeometric ODE, and without too much effort, as in [DJM14], we can obtain the solution $\phi(z) = \mu\left(z(1-z)\right)^{\alpha/2}$, for some constant μ and $\alpha = \alpha(\rho, \rho_0)$. We have thus

$$f(z_0, \ldots, z_{t-1}) = \nu_{s,t,K}\left(\prod_{i=0}^{t-1} z_i(1-z_i)\right)^{\alpha/2},$$

with $\nu_{s,t,K} := \mu^t$. This family of solutions (parameterized by the "arbitrary" $\nu_{s,t,K}$) obviously satisfies constraints (a), (b) and (c). Constraint (d) yields the sought function, as we impose

$$\nu_{s,t,K}\frac{\Gamma^{2t}(\alpha/2+1)}{\Gamma^t(\alpha+2)} = \beta(\rho, \rho_0).$$

C Bounding the errors

Once we have an explicit form for $f(\mathbf{z}) := f(z_0, \ldots, z_{t-1})$, we can compute error bounds for the successive approximations that led us from the recurrence in (7) to the integral equation (8). Our knowledge of the function $f(z_0, \ldots, z_{t-1})$ and its derivatives in $(0, 1)$ is the key to find these bounds. First, we can use the trapezoid rule or the Euler-Maclaurin summation formula to bound the error in

passing from sums to integrals; for instance

$$\frac{1}{n}\sum_{j=r_i}^{n-1} f\left(\frac{r_0}{n},\dots,\frac{r_i}{j},\dots,\frac{r_{t-1}}{n}\right)\left(\frac{j}{n}\right)^{\gamma} = \frac{1}{n}\int_{r_i}^{n} f\left(\frac{r_0}{n},\dots,\frac{r_i}{u},\dots,\frac{r_{t-1}}{n}\right)\left(\frac{u}{n}\right)^{\gamma} du - \frac{1}{2n} f\left(\frac{r_0}{n},\dots,\frac{r_i}{n},\dots,\frac{r_{t-1}}{n}\right) + O(n^{-2}),$$

and similarly for the other integrals.

Now, if we compare recurrence (7) for $P_{n,\mathbf{r}}$ to the recurrence (13) for $C_{n,\mathbf{r}}$, apart from the normalizing factor n^{γ}, the difference comes from the splitting probabilities $\pi_L^{(i,j)}(\mathbf{r},\mathbf{r}')$, $\pi_R^{(i,j)}(\mathbf{r},\mathbf{r}')$ and $\pi_B^{(i,j)}(\mathbf{r},\mathbf{r}',\mathbf{r}'')$, which we argued are highly concentrated around their respective means. Here, Laplace's method for summations can be used to bound the error in that step. For instance, take

$$\sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}} \pi_L^{(i,j)}(\mathbf{r},\mathbf{r}') f\left(\frac{r_0'}{j},\dots,\frac{r_i}{j},\dots,\frac{r_{t-1}'}{j}\right) = \nu_{s,t,K} \sum_{\mathbf{r}'\in\mathcal{L}_{\mathbf{r}}^{(i,j)}} \phi\left(\frac{r_i}{j}\right)\cdot\left(\prod_{\substack{0\le k<t\\ k\neq i}} \frac{\binom{j}{r_k'}\binom{n-j}{r_k-r_k'}}{\binom{n}{r_k}}\phi\left(\frac{r_k'}{j}\right)\right),$$

for some j such that $j/n\to c$, for some constant $0<c<1$. Now, the right-hand side above can be re-written as

$$\phi\left(\frac{r_i}{j}\right)\prod_{\substack{0\le k<t\\ k\neq i}}\sum_{0\le r_k'\le r_k}\frac{\binom{j}{r_k'}\binom{n-j}{r_k-r_k'}}{\binom{n}{r_k}}\phi\left(\frac{r_k'}{j}\right)$$

and we can deal with each factor separately (here, the fact that $f(z_0,\dots,z_{t-1}) = \phi(z_0)\cdots\phi(z_{t-1})$ greatly simplifies the proof). With our assumption that $r_k/n\to z_k$ for some $0<z_k<1$, we need just to show that

$$\sum_{0\le r_k'\le r_k}\frac{\binom{j}{r_k'}\binom{n-j}{r_k-r_k'}}{\binom{n}{r_k}}\phi\left(\frac{r_k'}{j}\right)\sim\phi\left(\frac{r_k}{n}\right)$$

The splitting probabilities are given by products of the hypergeometric distribution (owing to the independence with which coordinates of each data point are drawn)

$$\pi_L^{(i,j)}(k) = \frac{\binom{j}{r_k'}\binom{n-j}{r_k-r_k'}}{\binom{n}{r_k}}$$

and then we can apply the following approximation to the binomial distribution [JKK92] as long as $r_k = z_k n + o(n)$ and $j = cn + o(n)$

$$\frac{\binom{j}{r'_k}\binom{n-j}{r_k-r'_k}}{\binom{n}{r_k}} = \binom{r_k}{r'_k}\left(\frac{j}{n}\right)^{r'_k}\left(1-\frac{j}{n}\right)^{r_k-r'_k}\left(1+\frac{r'_k-(r'_k-\overline{r}_k)^2}{2j}+O\left(\frac{1}{j^2}\right)\right),$$

where $\overline{r}_k = r_k\frac{j}{n}$ is the mean value of the hypergeometric distribution.

If we divide the range of summation of r'_k into three parts, from 0 to $\overline{r}_k-\Delta-1$, from $\overline{r}_k-\Delta$ to $\overline{r}_k+\Delta$ and from $\overline{r}_k+\Delta+1$ to r_k, we can consider the three parts separately, with the main contribution coming from the middle range. In particular, we need $\Delta^3/n^2 \to 0$ as $n\to\infty$, that is $\Delta = o(n^{2/3})$, to be able to apply the de Moivre-Laplace limit theorem to the middle sum. With $\sigma = r_k\frac{j}{n}\left(1-\frac{j}{n}\right)$, we have that the middle sum is

$$\sum_{r'_k=\overline{r}_k-\Delta}^{\overline{r}_k+\Delta}\frac{1}{\sqrt{2\pi\sigma}}e^{-(r'_k-\overline{r}_k)^2/(2\sigma)+O(\Delta^3/n^2)+O(\Delta/n)} \times\phi\left(\frac{r'_k}{j}\right)\left(1+O(\Delta/n)+O(\Delta^2/n)+O\left(\frac{1}{n^2}\right)\right),$$

where we have also expressed the error bounds for the approximation of the hypergeometric distribution in terms of Δ; we need $\Delta = o(\sqrt{n})$ too for the approximation to be of any use. Using $e^x = 1+O(x)$ we can write the sum above as

$$\sum_{r'_k=\overline{r}_k-\Delta}^{\overline{r}_k+\Delta}\frac{1}{\sqrt{2\pi\sigma}}e^{-(r'_k-\overline{r}_k)^2/(2\sigma)}(1+O(\Delta/n))\phi\left(\frac{r'_k}{j}\right)\left(1+O(\Delta^2/n)\right),$$

since $O(\Delta^3/n^2) = O(\Delta/n)$ for $\Delta = o(\sqrt{n})$. Finally, we can expand $\phi(r'_k/j) = \phi(\overline{r}_k/j + y/j)$ for $y = r'_k-\overline{r}_k$, $y\in[-\Delta,\Delta]$ as $\phi(r'_k/j) = \phi(\overline{r}_k/j)+O(\Delta/n)$ to get

$$\sum_{r'_k=\overline{r}_k-\Delta}^{\overline{r}_k+\Delta}\frac{1}{\sqrt{2\pi\sigma}}e^{-(r'_k-\overline{r}_k)^2/(2\sigma)}(1+O(\Delta/n))\phi\left(\frac{\overline{r}_k}{j}\right)(1+O(\Delta/n))\left(1+O(\Delta^2/n)\right) = \phi\left(\frac{\overline{r}_k}{j}\right)(1+O(\Delta^2/n)) = \phi\left(\frac{\overline{r}_k}{j}\right)(1+o(1)).$$

To complete this part of the analysis we only need to show that the other two sums (with $r'_k < \overline{r}_k-\Delta$ and $r'_k > \overline{r}_k+\Delta$) are negligible as $n\to\infty$. This immediately follows since $\phi(r'_k/j)$ is bounded by a constant, and we only need to note that the tails of the hypergeometric distribution (or its binomial approximation) decay polynomially as we move away from the mean $\overline{r}_k$, then exponentially. To have an error bound as small as possible it helps to take Δ as large as possible, as long as it remains $o(\sqrt{n})$.

We handle the other inner sums (for rank vectors in $\mathcal{L}_{\mathbf{r}}^{(i,j)}$, $\mathcal{R}_{\mathbf{r}}^{(i,j)}$ and $\mathcal{B}_{\mathbf{r}}^{(i,j)}$) in (7) analogously; we have thus that the error bound inside each summation on j is $(1 + O(\Delta^2/n))$, but the approximations are not valid if $j = o(n)$. However these can be disregarded as their total contribution is negligible, since the tail of the hypergeometric distribution decays exponentially. This also justifies the assumption that all extreme ranks are $r_i = 0$ when we actually have extreme ranks $r_i = o(n)$ (or $r_i = n - o(n)$, but then we can take $r_i := n - r_i$ because of the symmetry).

Altogether, these computations show that $C'_{n,\mathbf{r}} = f(\mathbf{r}/n) + o(1)$ satisfies

$$C'_{n,\mathbf{r}} \sim o(1) + \frac{1}{nK}\left[\sum_{0\le i<t}\left(\sum_{j=r_i}^{n-1} C'_{j,\overleftarrow{\mathbf{r}}}\cdot\left(\frac{j}{n}\right)^{\gamma} + \sum_{j=0}^{r_i-1} C'_{n-1-j,\overrightarrow{\mathbf{r}}}\cdot\left(\frac{n-1-j}{n}\right)^{\gamma}\right)\right] \tag{15}$$

$$+ \frac{s_0}{nK}\cdot\sum_{j=0}^{n-1} C'_{j,\overleftarrow{\mathbf{r}}}\cdot\left(\frac{j}{n}\right)^{\gamma}$$

$$+ \frac{(K-s)}{nK}\cdot\sum_{j=0}^{n-1}\left(C'_{j,\overleftrightarrow{\mathbf{r}}}\cdot\left(\frac{j}{n}\right)^{\gamma} + C'_{n-1-j,\mathbf{r}-\overleftrightarrow{\mathbf{r}}}\cdot\left(\frac{n-1-j}{n}\right)^{\gamma}\right),$$

and hence $f(\mathbf{r}/n)\cdot n^{\alpha} + o(n^{\alpha})$ satisfies the full recurrence (7), with toll function $o(n^{\alpha})$ instead of the toll function $\tau_{n,\mathbf{r}} = 1$.